GigaDevice Semiconductor Inc.

GDSCN832xx

User Manual

Revision 1.1

(May. 2025)

Table of Contents

List of Figures

List of Tables

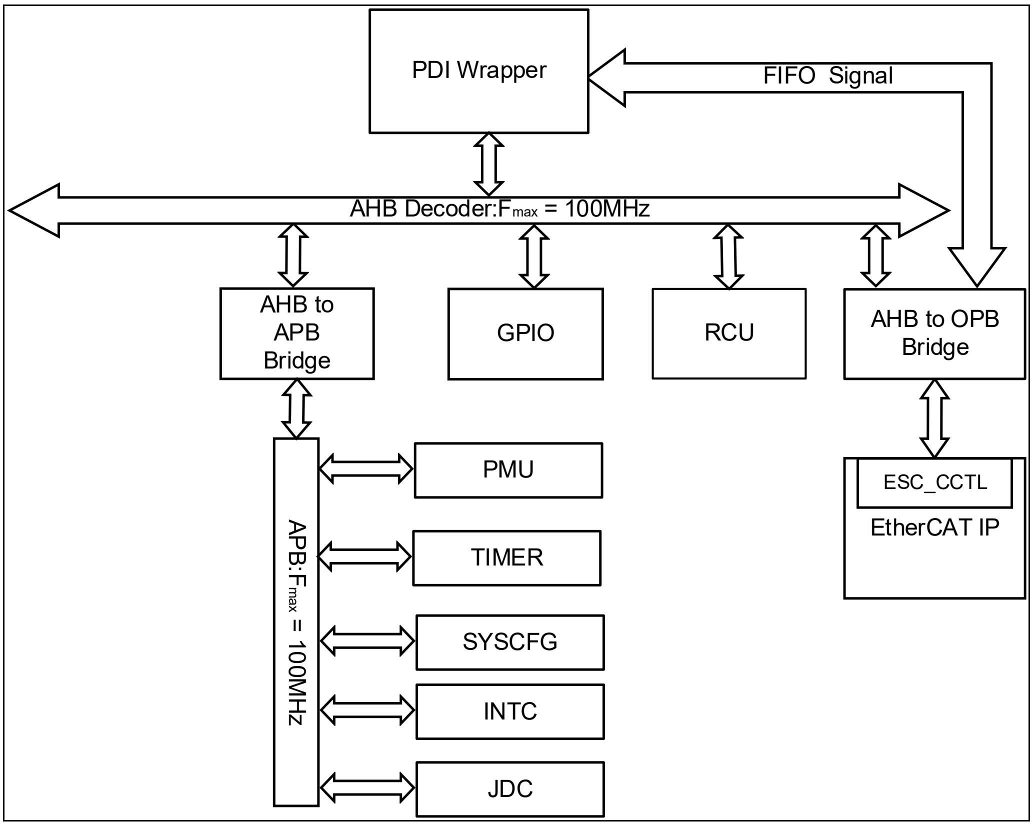
1. System and bus architecture

1.1. Bus architecture

The bus architecture of ESC is shown in the following figure. The AHB matrix based on AMBA 5 AHB-LITE is a multi-layer AHB, which enables parallel access paths between one masters and multiple slaves in the system. One masters on the AHB decoder, including AHB bus of the PDI Wrapper. The AHB decoder consists of five slaves, including the AHB to OPB Bridge, AHB to APB Bridge, GPIO and RCU.

The AHB connects with the AHB peripherals including one AHB-to-APB bridges which provide full synchronous connections between the AHB decoder and the one APB buses. The one APB buses connect with all the APB peripherals, including the PMU, TIMER, SYSCFG, INTC.

Figure 1-1. Bus architecture of ESC



1.2. Memory map

This section will introduce the memory distribution of GDSCN as shown in ***Table 1-1. Memory map of GDSCN*** below.

Table 1-1. Memory map of GDSCN

Pre-defined Regions	Address	Peripherals
EtherCAT	0x0000	Type
	0x0001	Revision
	0x0002 - 0x0003	Build
	0x0004	FMMUs Supported
	0x0005	SyncManagers Supported
	0x0006	RAM Size
	0x0007	Port Descriptor
	0x0008 - 0x0009	ESC Features Supported
	0x0010 - 0x0011	Configured Station Address
	0x0012 - 0x0013	Configured Station Alias
	0x0013 - 0x001F	Reserved
	0x0020	Write Register Enable
	0x0021	Write Register Protection
	0x0022 - 0x002F	Reserved
	0x0030	ESC Write Enable
	0x0031	ESC Write Protection
	0x0032 - 0x003F	Reserved
	0x0040	ESC Reset ECAT
	0x0041	ESC Reset PDI
	0x0042 - 0x0FF	Reserved
	0x0100 - 0x0103	ESC DL Control
	0x0104 - 0x0107	Reserved
	0x0108 - 0x0109	Physical Read/Write Offset
	0x0110 - 0x0111	ESC DL Status
	0x0112 - 0x011F	Reserved
	0x0120 - 0x0121	AL Control
	0x0122 - 0x012F	Reserved
	0x0130 - 0x0131	AL Status
	0x0132 - 0x0133	Reserved
	0x0134 - 0x0135	AL Status Code
	0x0136 - 0x0137	Reserved
	0x0138	RUN LED Override
	0x0139	Reserved
	0x0140	PDI Control
	0x0141	ESC Configuration
	0x0142 - 0x0143	ASIC Configuration
	0x0144 - 0x0145	RESERVED Register
	0x0146 - 0x014F	Reserved
	0x0150	PDI Configuration
	0x0151	Sync/Latch PDI Configuration

Pre-defined Regions	Address	Peripherals
	0x0152 - 0x0153	Extended PDI Configuration
	0x0154 - 0x01FF	Reserved
	0x0200 - 0x0201	ECAT Event Mask
	0x0202 - 0x0203	Reserved
	0x0204 - 0x0207	PDI AL Event Mask
	0x0208 - 0x0209	Reserved
	0x0210 - 0x0211	ECAT Event Request
	0x0212 - 0x021F	Reserved
	0x0220 - 0x0223	AL Event Request
	0x0223 - 0x022F	Reserved
	0x0300 - 0x0307	RX Error Counter
	0x0308 - 0x030B	Forwarded RX Error Counter
	0x030C	ECAT Processing Unit Error Counter
	0x030D	PDI Error Counter
	0x030E	PDI Error Code
	0x030F	Reserved
	0x0310 - 0x0313	Lost Link Counter
	0x0314 - 0x03FF	Reserved
	0x0400 - 0x0401	Watchdog Divider
	0x0410 - 0x0411	Watchdog Time PDI
	0x0420 - 0x0421	Watchdog Time Process Data
	0x0440 - 0x0441	Watchdog Status Process Data
	0x0442	Watchdog Counter Process Data
	0x0443	Watchdog Counter PDI
	0x0444 - 0x04FF	Reserved
	0x0500	EEPROM Configuration
	0x0501	EEPROM PDI Access State
	0x0502 - 0x0503	EEPROM Control/Status
	0x0504 - 0x0507	EEPROM Address
	0x0508 - 0x050B	EEPROM Data
	0x050C- 0x050F	Reserved
	0x0510 - 0x0511	MII Management Control/Status
	0x0512	PHY Address
	0x0513	PHY Register Address
	0x0514 - 0x0515	PHY DATA
	0x0516	MII Management ECAT Access State
	0x0517	MII Management PDI Access State
	0x0518 - 0x051B	PHY Port Status
	0x051C - 0x05FF	Reserved
	0x0600 - 0x06FF	FMMU
	0x0700 - 0x07FF	Reserved

Pre-defined Regions	Address	Peripherals
	0x0800 - 0x087F	SyncManager
	0x0880 - 0x08FF	Reserved
	0x0900 - 0x09FF	Distributed Clocks (DC)
	0x0A00 - 0x0AFF	Reserved
	0x0E00 - 0x0E07	Product ID
	0x0E08 - 0x0E0F	Vendor ID
	0x0E10 - 0x0EFF	Reserved
	0x0F00 - 0x0F03	Digital I/O Output Data
	0x0F04 - 0x0F0F	Reserved
	0x0F10 - 0x0F17	General Purpose Outputs
	0x0F18 - 0x0F1F	General Purpose Inputs
	0x0F20 - 0x0F7F	Reserved
	0x0F80 - 0x0FFF	User RAM
	0x1000 - 0x2FFF	Process Data RAM
Peripheral	0x3300 - 0x33FF	AHB2OPB Bridge
	0x3400 - 0x34FF	RCU
	0x3500 - 0x35FF	GPIO
	0x3700 - 0x37FF	PMU
	0x3800 - 0x38FF	TIMER
	0x3900 - 0x39FF	SYSCFG
	0x3A00 - 0x3AFF	INTC

1.3. AHB direct/indirect access

GDSCN can directly/indirectly access ESC registers and core PRAM through the AHB to OPB bridge in three ways: directly accessible ESC registers, indirectly accessible ESC core registers, and indirectly accessible ESC core PRAM. The bridge provides an AHB Slave interface towards the upstream side and OPB interfaces towards the downstream side of the ESC core, where the upstream AHB side is faster than, in frequency to, the downstream ESC core, and the clocks are synchronous in phase, and have an N:1(max N=16) frequency ratio.

1.3.1. Direct AHB transmit access ESC register

Direct access to ESC registers is used for transferring data/commands to the indirect access of ESC core registers. When the AHB bus accesses the address range of 0x0000-0x0FFF, a single register read/write operation on the ESC core is initiated based on the CCTL_RW bit in the ESC_CCTL_CMD register. At the start of a read cycle, the CCTL_BUSY bit in the ESC_CCTL_CMD register is set to 1, and at the end of the read cycle, CCTL_BUSY is cleared to 0, allowing valid data to be read on the AHB bus. At the start of a write cycle, the CCTL_BUSY bit is set to 1, valid data is written to the bus, and at the end of the write cycle, the CCTL_BUSY bit is cleared, and the valid data is written into the register.

1.3.2. Indirect transmit access ESC core register

GDSCN can access the ESC core registers indirectly through the ESC_CCTL_DATA and ESC_CCTL_CMD registers. When reading an ESC core register, the following steps are required: First, set the CCTL_STOP bit in the ESC_CCTL_CMD register to 1 to clear the CCTL_BUSY bit; set the CCTL_RW bit in the ESC_CCTL_CMD register to 1, write the address of the register to be accessed into the CCTL_ADDR field, and write the number of bytes to be read into the CCTL_SIZE field, then set the CCTL_BUSY bit to 1. When the CCTL_BUSY bit is cleared, data can be read from the ESC_CCTL_DATA register.

When writing data to the ESC core registers, the following steps are required: First, set the CCTL_STOP bit in the ESC_CCTL_CMD register to 1 to clear the CCTL_BUSY bit; clear the CCTL_RW bit in the ESC_CCTL_CMD register, write the address of the register to be accessed into the CCTL_ADDR field, and write the number of bytes to be written into the CCTL_SIZE field, then set the CCTL_BUSY bit to 1. The configuration data can be written into the ESC_CCTL_CMD register at once. The completion of the write cycle is indicated by the CCTL_BUSY bit being cleared to zero.

In the above read/write operations, the valid data is always aligned with the low bits of the ESC_CCTL_DATA. The valid data can be referred to the following table.

Table 1-2. Alignment of valid data

CCTL_SIZE	ESC CCTL_ADDR[1:0]	ESC CCTL_DATA valid bytes
1	00/01/10/11	[7:0]/ [15:8]/ [23:16]/ [31:24]/
2	00/10	[15:0]/[31:16]
4	00	[31:0]

1.3.3. Indirect transmit access ESC core PRAM

When initiating read operation to core PRAM through AHB, after writing PRAM start address and read length to ESC_PRAM_ALR register, PRAM_BUSY_READ is set to 1, the module starts to initiate multiple OPB read operations, read data from core PRAM and write to the TX FIFO. All OPB read operations are complete. PRAM_BUSY_READ is cleared. When data is transferred from the ESC core to the TX FIFO, the PRAM read length PRAM_LEN_READ and the PRAM read address PRAM_ADDR_READ are updated to show the process. Determines the valid bytes of the first read data according to the start address. Determines the valid bytes of the last read data based on the starting address and operation length. If necessary, the read command can be stopped by setting the ESC_PRAM_CR[PRAM_STOP_READ] bit to 1. If the OPB read period starts, the stop command takes effect after the current read operation is complete. After the stop command takes effect, data in the TX FIFO is cleared.

When writing to the PRAM through AHB, after writing the RAM start address and write length to the ESC_PRAM_ALW register, write 1 to the PRAM_BUSY_WRITE bit of the ESC_PRAM_CW register, the module initiates multiple OPB write operations. Data is read from the RX FIFO and written to the core PRAM. All OPB write operations are complete. PRAM_BUSY_WRITE is cleared Write operations support wait mechanism, PRAM_BUSY_WRITE bit is set to 1, but when RX FIFO is empty, OPB module will not immediately initiate OPB transfer operation, until there is data in RX FIFO will initiate this operation. After each OPB ACK response, it will detect the RX FIFO state. If the RX FIFO is empty, it will enter the waiting state and enter the next transmission operation until there is data. When data is transferred from the RX FIFO to the core, the PRAM write length PRAM_LEN_WRITE and the PRAM write address PRAM_ADDR_WRITE are updated to show the process. Determines the valid bytes of the first data write according to the start address. Based on the starting address and operation length, determines the valid bytes of the last data write.

If necessary, the write command can be stopped by setting the PRAM_STOP_WRITE bit to 1. If the OPB write cycle starts, the stop command takes effect after the read operation is complete. After the stop command takes effect, data in the RX FIFO is cleared.

1.4. Register protection in BUSY state

If the BUSY related register bits, such as CCTL_BUSY, PRAM_BUSY_READ, and PRAM_BUSY_WRITE, are set to 1, you can set the BRP bit to protect the register from rewriting. When CCTL_BUSY is set to 1, ESC_CCTL_DATA/ ESC_CCTL_CMD is protected from rewriting by AHB write operations. When PRAM_BUSY_READ is set to 1, ESC_PRAM_ALR is protected from rewriting by AHB write operations. When PRAM_BUSY_WRITE is set to 1, ESC_PRAM_ALW is protected from being overwritten by AHB write operations

When the BRP bit is set to 1, when the BUSY related register bit is set to 1, the user's AHB write operation to the corresponding register will be lost, then the WDLF flag in the ESC_OPB_CS register is set to 1, and the interrupt will be triggered when the WDIE bit is set to 1.

When the BRP bit is set to 0, when the BUSY related register bit is set to 1, the user's AHB write operation to the corresponding register will cause the current OPB transmission error, then the WEF flag in the ESC_OPB_CS register is set to 1, and the interrupt will be triggered when the WEIE bit is set to 1. It is advised to handle it as soon as possible to avoid more errors.

1.5. OPB transmission timeout function

The OPB transmission timeout function can be enabled by setting the TOEN bit in the ESC_OPB_CS register to 1. The timeout interval can be configured by the TO_CNT bit in the ESC_OPB_CS register. When the counter exceeds the TO_CNT programming value, the TOF flag in the ESC_OPB_CS register is set to 1. If the TOIE bit in the ESC_OPB_CS register is set to 1, a timeout interrupt is generated. Timeout interrupt response processing:

1. If the ESC CCTL direct read/write mode is used, the TOF flag is set after a timeout

interrupt response occurs. Terminate the transmission directly to avoid the BUS being occupied.

2. If the ESC CCTL is used in indirect read/write mode, the TOF flag is set after a timeout interrupt response occurs. You can stop this operation by writing 1 to the ESC CCTL_STOP bit.

3. When the PRAM is used in indirect read/write mode, the TOF flag is set after a timeout interrupt response occurs. You can stop this operation by writing 1 to the PRAM_STOP_WRITE/ PRAM_STOP_READ bit.

1.6. ESC core controller (ESC_CCTL)

The main purposes of the ESC core controller (ESC_CCTL) are the following:

- Configuring indirect transmit access of the ESC core register
- Configuring indirect transmit access of the ESC core PRAM.

1.7. ESC core controller register definition

ESC core control register base address: 0x3300

1.7.1. ESC CCTL data register (ESC_CCTL_DATA)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">CCTL_DATA[31:16]</td></tr>
<tr><td colspan="16">rw</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">CCTL_DATA[15:0]</td></tr>
<tr><td colspan="16">rw</td></tr>
</table>

Bits	Fields	Descriptions
31:0	CCTL_DATA[31:0]	ESC CCTL data This field indicate the value read from or written to the ESC Core. Reading or writing depends on the ESC_CCTL_CMD[CCTL_RW] bit. If the CCTL_RW bit is 1, this value is the data read from the ESC Core; If the CCTL_RW bit is 0, this value is the data written to the ESC Core. The low bit of this filed always indicates the valid data written or read.

1.7.2. ESC CCTL command register (ESC_CCTL_CMD)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCTL_BUSY	CCTL_RW	CCTL_STOP	Reserved										CCTL_SIZE[2:0]		
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCTL_ADDR[15:0]															
rw															

Bits	Fields	Descriptions
31	CCTL_BUSY	CCTL Busy. There is no effect if the bit is written to zero 0: No read/write (according to CCTL_RW) operation is being performed. 1: Read/write operation is being performed. **Note:** When the read/write operation is complete, this bit will be cleared. Then the valid data is available for the HOST to read from ESC_CCTL_DATA or write to ESC_CCTL_DATA register. It is required that ESC_CCTL_CMD and ESC_CCTL_DATA registers are modified R when this bit is 0.
30	CCTL_RW	Read operation or write operation 0: Write operation 1: Read operation
29	CCTL_STOP	Stop read operation or write operation. There is no effect if the bit is written to zero 0: No effect 1: Stop the ESC core register reading or writing operation **Note:** When the CCTL_BUSY is cleared, the CCTL_STOP will be cleared
28:19	Reserved	Must be kept at reset value.
18:16	CCTL_SIZE[2:0]	This field specifies the ESC CCTL size (byte). 1, 2 and 4 are valid, other values are invalid. More details refer to ***Table 1-2. Alignment of valid data***
15:0	CCTL_ADDR[15:0]	This field specifies the addresses of ESC core registers that will be accessed

1.7.3. ESC PRAM FIFO data read register (ESC_PRAM_FIFO_DR)

Address offset: 0x10

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_FIFO_DATA_READ[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_FIFO_DATA_READ[15:0]															
rw															

Bits	Fields	Descriptions
31:0	PRAM_FIFO_DATA_READ[31:0]	Data read from ESC PRAM. The valid value of data is determined according to the start address and the transfer length.

1.7.4. ESC PRAM address and length read register (ESC_PRAM_ALR)

Address offset: 0x14

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_LEN_READ[15: 0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_ADDR_READ [15:0]															
rw															

Bits	Fields	Descriptions
31:16	PRAM_LEN_READ[15: 0]	Data length read from ESC PRAM in bytes. This field is decremented as data is read into the FIFO. **Note:** When PRAM_BUSY_READ is 1, this field cannot be modified.
15:0	PRAM_ADDR_READ [15:0]	ESC PRAM data read address. This field is incremented as data is read into the FIFO. **Note:** When PRAM_BUSY_READ is 1, this field cannot be modified.

1.7.5. ESC PRAM command read register (ESC_PRAM_CR)

Address offset: 0x18

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_BUSY_READ	PRAM_STOP_READ	Reserved													
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PRAM_VALID_CNT_READ[4:0]					Reserved							PRAM_VALID_READ

r r

Bits	Fields	Descriptions
31	PRAM_BUSY_READ	Whether the PRAM is being read. There is no effect if the bit is written to zero 0: No PRAM read operation 1: PRAM is being read **Note:** This bit is cleared when the read operation is complete.
30	PRAM_STOP_READ	Stop PRAM read operation. There is no effect if the bit is written to zero 0: No effect 1: Stop the PRAM reading operation Note: After this bit is set to 1, the PRAM_BUSY_READ is cleared, and the RX FIFO is reset. Then this bit will self-clear.
29:13	Reserved	Must be kept at reset value.
12:8	PRAM_DATA_CNT_READ[4:0]	PRAM data read valid count This count increases as data is read from the PRAM to the RX FIFO and decreases as data of the entire DWORD size is read from the RX FIFO
7:1	Reserved	Must be kept at reset value.
0	PRAM_VALID_DATA_READ	PRAM valid data read 0: There is no valid data to be read. 1: There is valid data to be read.

1.7.6. ESC PRAM FIFO data write register (ESC_PRAM_FIFO_DW)

Address offset: 0x20

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_FIFO_DATA_WRITE[31:16]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_FIFO_DATA_WRITE[15:0]															
rw															

Bits	Fields	Descriptions
31:0	PRAM_FIFO_DATA_WRITE[31:0]	Data write to ESC PRAM. The valid value of data is determined according to the start address and the transfer length.

1.7.7. ESC PRAM address and length write register (ESC_PRAM_ALW)

Address offset: 0x24

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_LEN_WRITE[15: 0]															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRAM_ADDR_WRITE[15:0]															
rw															

Bits	Fields	Descriptions
31:16	PRAM_LEN_WRITE[15: 0]	Data length written to ESC PRAM in bytes. This field is decremented as data is written into the FIFO. **Note:** When PRAM_BUSY_WRITE is 1, this field cannot be modified.
15:0	PRAM_ADDR_WRITE[15:0]	ESC PRAM data write address. This field is incremented as data is read from the FIFO. **Note:** When PRAM_BUSY_ WRITE is 1, this field cannot be modified.

1.7.8. ESC PRAM command write register (ESC_PRAM_CW)

Address offset: 0x28

Reset value: 0x0000 1001

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRAM_BUSY_WRITE	PRAM_STOP_WRITE	Reserved													
rw	rw														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PRAM_VALID_CNT_WRITE[4:0]					Reserved							PRAM_VALID_WRITE
			r												r

Bits	Fields	Descriptions
31	PRAM_BUSY_WRITE	Whether the PRAM is being written. There is no effect if the bit is written to zero 0: No PRAM write operation 1: PRAM is being writing **Note:** This bit is cleared when the write operation is complete.
30	PRAM_STOP_WRITE	Stop PRAM write operation. There is no effect if the bit is written to zero 0: No effect 1: Stop the PRAM writing operation Note: After this bit is set to 1, the PRAM_BUSY_WRITE is cleared, and the TX

		FIFO is reset. Then this bit will self-clear.
29:13	Reserved	Must be kept at reset value.
12:8	PRAM_VALID_CNT_WRITE[4:0]	PRAM data write valid count This count increases as data is read from the TX FIFO to PRAM and decreases as data of the entire DWORD size is written to the PRAM
7:1	Reserved	Must be kept at reset value.
0	PRAM_VALID_DATA_WRITE	PRAM valid data write 0: There is no valid data to be written 1: There is valid data to be written.

1.7.9. ESC OPB control and status register (ESC_OPB_CS)

Address offset: 0x30

Reset value: 0x0000 04A0

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RAAF	RAAIE	WDLF	WDLIE	TOF	TOIE	WEF	WEIE	ESC CCTLIVF	ESC CCTLIVIE	Reserved					
rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw	rc_w1	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					BRP	TO_CNT[8:0]									TOEN
					rw	rw									rw

Bits	Fields	Descriptions
31	RAAF	Reserved address access flag. This flag will be triggered when the access address is a reserved address segment. This flag can be cleared by writing 1. 0: The access address is a valid address segment 1: The access address is a reserved address segment
30	RAAIE	Reserved address access interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the RAAF bit is set
29	WDLF	Write data lost flag. when BRP enable, AHB write operation will cause data lose when the busy bit is high, this flag will be triggered. This flag can be cleared by writing 1. 0: No data lost 1: The data in this transmission is lost
28	WDLIE	Write data lost interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the RAAF bit is set

27	TOF	Time out flag, when single transmission time exceeds the programmed value, this flag will be triggered. This flag can be cleared by writing 1. 0: No ESC core timeout transmission occurred 1: An ESC core timeout transmission occurred
26	TOIE	Timer out interrupt enable 0: Interrupt is inhibited 1: An interrupt will occur whenever the RAAF bit is set
25	WEF	write error flag. when BRP disable, AHB write operation will cause current OPB transmission error when the busy bit is high, this flag will be triggered. This flag can be cleared by writing 1.
24	WEIE	Write error interrupt enable. 0: Interrupt is inhibited 1: An interrupt will occur whenever the WEF bit is set
23	ESC CCTLIVF	CCTL_SIZE & CCTL_ADDR illegal value flag. The CCTL_SIZE and CCTL_ADDR value written to the register does not meet the requirement, causing this transfer to not occur, this flag will be triggered. This flag can be cleared by writing 1.
22	ESC CCTLIVIE	CCTL_SIZE & CCTL_ADDR Illegal value interrupt enable. 0: Interrupt is inhibited 1: An interrupt will occur whenever the IESC CCTLVF bit is set
21:11	Reserved	Must be kept at reset value.
10	BRP	When busy bit is high, prevent register change 1: the protection takes effect 0: the protection does not take effect
9:1	TO_CNT[8:0]	Time Out counter This field indicates the transmission timeout in bits during which no ACK response was received; The programmer data cannot be less than the minimum value recommended the manual(60).
0	TOEN	Time Out enable This field indicates that whether the function is effect 1: Timeout feature enabled 0: Timeout feature disabled.

1.8. System configuration controller (SYSCFG)

The main purposes of the system configuration controller (SYSCFG) are the following:

- Configuring MCU HCLK frequency ratio.
- Configuring SPI extend mode.
- Providing Chip ID and version.

1.9. System configuration register definition

SYSCFG base address: 0x000 3900

1.9.1. System configuration register 0 (SYSCFG_CFG0)

Address offset: 0x00
Reset value: 0x001F 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											MCUFREQ[4:0]				
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SPIEXTMOD	Reserved	
													rw		

Bits	Fields	Descriptions
31:21	Reserved	Must be kept at reset value.
20:16	MCUFREQ[4:0]	MCU HCLK frequency ratio 00000: MCU_HCLK_FREQ >=100MHZ 00001: 100MHZ/2 <= MCU_HCLK_FREQ < 100MHZ 00010: 100MHZ/3 <= MCU_HCLK_FREQ < 100MHZ/2 ... 11111: 100MHZ/32 <= MCU_HCLK_FREQ < 100MHZ/31
15:3	Reserved	Must be kept at reset value.
2	SPIEXTMOD	PDI type combined with SPI 0: GPIO 1: MII (OSPI: without clk_25m output; Others: with clk_25M)
1:0	Reserved	Must be kept at reset value.

1.9.2. SYSCFG chip id register (SYSCFG_CHIPID)

Address offset: 0x90
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHIPID[15:0]															
							r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved

Bits	Fields	Descriptions
31:16	CHIPID[15:0]	Chip ID
15:0	Reserved	Must be kept at reset value

1.9.3. SYSCFG chip version register (SYSCFG_CHIPVER)

Address offset: 0x94

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CHIPVER[7:0]							
								r							

Bits	Fields	Descriptions
31:8	Reserved	Must be kept at reset value.
7:0	CHIPVER[7:0]	Chip version

1.9.4. SYSCFG reserved register (SYSCFG_RESERVED)

Address offset: 0xF0

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															
rw															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	RESERVED	Reserved register

2. Power management unit (PMU)

2.1. Overview

The power consumption is regarded as one of the most important issues for the devices of ESC series. Power management unit (PMU) provides four types of device level and three types of module level power saving modes, device level power saving modes including MOD0, MOD1, MOD2 and MOD3, module level power saving modes including EtherCAT clock management, PHY power management and the LED pins power management. These modes reduce the power consumption and allow the application to achieve the best tradeoff among the conflicting demands of device operating time, speed and power consumption. PMU also supports wake up event detection and power management event (PME) notification.

2.2. Characteristics

- EtherCAT clock management.
- PHY power management, including PHY A and B energy detect(ED) power down management and common power down management.
- LED pins power down management.
- Four types of device level power saving modes, including MOD0, MOD1, MOD2 and MOD3.
- PHY wake up event detection, including PHY ED powered wake up and PHY LAN magic packet wake up.
- Interrupt wake up notification.

2.3. Function overview

2.3.1. Device ready

The bit RDY in PMU_CTL0 register, can indicate whether the device is ready. The master processor can read this bit to obtain the ready status of the device.

- After power on, EtherCAT device reset or digital reset, if RDY bit is set, it indicates that the device has successfully read the contents of the EEPROM and is configured according to the read contents.
- Setting the ESCRST bit in the RCU_RSTCFG register will reset the EtherCAT core, which will cause the EtherCAT core to re-read the EEPROM and reconfigure according to the configuration content. During this process, the RDY bit will momentarily transition to a low level.
- When the device enters power-saving modes MOD1, MOD2, or MOD3, the RDY bit will transition to a low level. Once the device is awakened from a power-saving mode back into the MOD0, and after the PLL becomes stable, the RDY bit will be set back to a high

level.

Note: The device supports voltage detection. The RDY bit can be set when the supply voltage reaches a predetermined value.

2.3.2. PHY wake up event detection

The device supports two types of PHY wake-up event detection:

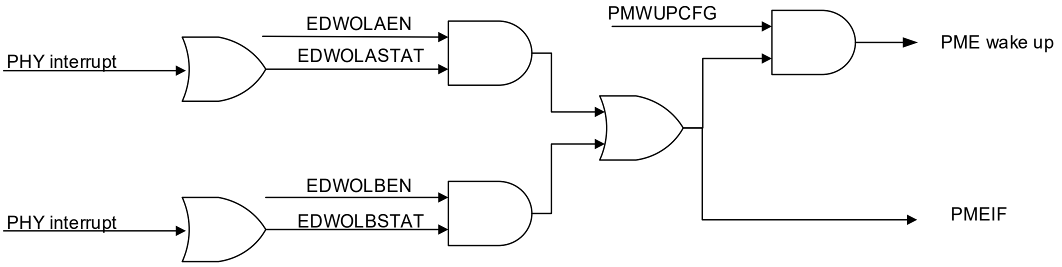
- PHY ED powered wake up event.
- PHY LAN magic packet wake up.

2.3.3. PME wake up notification

The latches of bit EDWOLASTAT and EDWOLBSTAT in the PMU_CTL register is handled in the PME module. Refer to the ***Figure 2-1. PME interrupt pending*** to understand the logic of PME interrupt control. If EDWOLAEN or EDWOLBEN is set, when energy detect / WoL event happened on port A or B PHY the PMEIF bit in interrupt status register will be set.

When PMWUPCFG is set, PME events can automatically wake up the system in some device-level power saving modes.

Figure 2-1. PME interrupt pending



2.3.4. Module level power saving modes

The device supports three types of module level power saving modes:

- The ECATCLKDIS bit in PMU_CTL0 register can use to disable EtherCAT core clock.
- PHY power management
  - PHY A and B ED power down management, support auto ED power down.
  - Common power down management.
- LEDs output management
  - The LEDOUTDIS bit in PMU_CTL0 register can use to disable LEDs output.
  - The LEDMODCFG bit in PMU_CTL0 register can use to configure LEDs working mode (take effect only when LEDOUTDIS is set).
  - The LEDINACT bit in PMU_CTL0 register can use to configure the inactive state when LEDs work in push-pull mode (take effect only when LEDOUTDIS is set).

2.3.5. Device level power saving modes

After a device level reset, the device operates at full function and all clocks are active. Users can achieve lower power consumption through gating the clocks of the unused functions. Besides, four device level power saving modes are provided to achieve even lower power consumption, they are MOD0, MOD1, MOD2 and MOD3.

MOD0

After a device level reset, the device works in MOD0 and operates at full function, all clocks are active.

MOD1

When in MOD1, device will disable all clocks derived from the PLL clock. If powered via PHY or externally, the network clock remains enabled. The XTAL and PLL remain enabled. This mode can be exited either manually or automatically.

This mode is applicable to the PHY's common power down management, PHY's WoL (Wake on LAN) mode, and PHY's ED power down management.

MOD2

When in MOD2, device will disable all clocks derived from the PLL clock. If powered via PHY or externally, the network clock remains enabled. It is allowed to disable the PLL (it will be disabled if both PHYs are in ED or common power down management). The XTAL and PLL remain enabled. This mode can be exited either manually or automatically.

This mode is applicable to the PHY's common power down management, PHY's WoL (Wake on LAN) mode, and PHY's ED power down management.

MOD3

When in MOD3, device will disable all clocks derived from the PLL clock. The PLL is disabled. The external network clocks are turned off. The crystal oscillator is disabled. This mode can only be exited manually.

This mode is applicable to the PHY's common power down management.

Before setting this power state, the master device should set POWERDOWN bit.

Table 2-1. Power saving mode summary

Mode	PLL	System clocks	Network clocks	XTAL
MOD0	ON	ON	usable	ON
MOD1	ON	OFF	usable	ON
MOD2	OFF	OFF	usable	ON
MOD3	OFF	OFF	OFF	OFF

2.3.6. Entering device level power saving modes

To transition from MOD0 to MOD1, MOD2, or MOD3, can follow these steps:

1. Configure PMWUPCFG bit.
2. Configure PHY wake up detection, about PHY wake up detection can refer to ***PHY wake up event*** detection.
3. Configure PHY wake up notification, about PHY wake up notification can refer to ***PME wake up notification***.
4. ensure that the device has been able to enter the power saving mode (ensure that there is no need to send data packets, receivers disabled, etc.).
5. Set PMSLPEN bit.

Note:

- After entering power saving mode, the RDY bit in registers PMU_CTL0 will be set to low level.
- After entering power saving mode, the master interface is invalid.

2.3.7. Exiting device level power saving modes

The device level power saving modes can be exited either manually or automatically.

If PMWUPCFG bit is set, the PME wake up is enabled, PME automatically wake up may occur. About PME wake up can refer to ***PME wake up notification***.

The master can manually wake up device by:

- Perform EXMC write operations on the device. Although all write operations are ignored until the device is woken up and a read operation performed, the master should still indicate a write to the PMU_PDIREFVAL register. No attempt should be made to write any other address until the device has been woken up.
- Perform SPI/SQI cycles on the device (CS low and SCK high). Although all read and write operations are ignored until the device is woken up, the master should still indicate to wake up the device by reading the PMU_PDIREFVAL register. No attempt should be made to read and write any other address until the device has been woken up.

Note:

- The working state of the master interface can be determined by reading the PMU_PDIREFVAL register. Once the correct value is read, the master interface will enter the ready state. Then the RDY bit will indicate when the device is fully awakened.
- After automatic or manual wakeups, the device RDY bit is set once the device has returned to MOD0 and the PLL has been re-stabilized, the PMMODCFG and PMSLPEN bits or bit will be cleared (set to 0).
- If all is well, the device wake up time should be less than 2 ms.

2.4. Register definition

PMU base address: 0x0000 3700

2.4.1. Control register 0 (PMU_CTL0)

Address offset: 0x00

Reset value: 0x0000 C000.

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMMODCFG[1:0]		Reserved		PMWUPC FG	LEDOUTD IS	LEDMOD CFG	LEDINAC T	Reserved		ECATCLK DIS	Reserved			EDWOLB STAT	EDWOLA STAT
rw				rw	rw	rw	rw			rw				r_w1	r_w1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EDWOLB EN	EDWOLA EN	Reserved													RDY
rw	rw														r

Bits	Fields	Descriptions
31:30	PMMODCFG[1:0]	Power management mode configuration bits When PMSLPEN is set, these bits can be used to configure the power management mode These bits are cleared when the device wakes up. 00: MOD0 01: MOD1 10: MOD2 11: MOD3
29:28	Reserved	Must be kept at reset value
27	PMWUPCFG	Power management wake up mode configuration bit 0: Wake up by master 1: Wake up by PME or master
26	LEDOUTDIS	LEDs output disabled bit When this bit is set, the output of LEDs will be disabled. When work in open-drain / open-source mode LEDs will not be driven, when LEDs work in push-pull mode LEDs will still be driven but the status of LEDs are inactive. 0: LEDs output are enabled 1: LEDs output are disabled
25	LEDMODCFG	LEDs working mode configuration bit (take effect only when LEDOUTDIS is set) 0: The working mode of LEDs is open-drain / open-source 1: The working mode of LEDs is push-pull
24	LEDINACT	Push-pull mode LEDs inactive state configuration bit (take effect only when

		LEDOUTDIS is set) 0: 0 is inactive state 1: 1 is inactive state
23:22	Reserved	Must be kept at reset value
21	ECATCLKDIS	EtherCAT core clock disable bit EtherCAT core clock will disable if ECATCLKDIS is set. To set this bit need write 1 twice in a row. Writing 0 will clear the count. 0: Enable EtherCAT core clock 1: Disable EtherCAT core clock
20:18	Reserved	Must be kept at reset value
17	EDWOLBSTAT	Energy detect / WoL port B status bit To clear this bit, the events on the PHY need to clear first. 0: No energy detect / WoL event happened on port B PHY 1: energy detect / WoL event happened on port B PHY
16	EDWOLASTAT	Energy detect / WoL port A status bit To clear this bit, the events on the PHY need to clear first. 0: No energy detect / WoL event happened on port A PHY 1: energy detect / WoL event happened on port A PHY
15	EDWOLBEN	Energy detect / WoL port B enable bit When energy detect / WoL event happened on port B PHY and this bit is enabled, the PMEIF bit in interrupt status register will be set. 0: Disable energy detect / WoL port B 1: Enable energy detect / WoL port B
14	EDWOLAEN	Energy detect / WoL port A enable bit When energy detect / WoL event happened on port A PHY and this bit is enabled, the PMEIF bit in interrupt status register will be set. 0: Disable energy detect / WoL port A 1: Enable energy detect / WoL port A
13:1	Reserved	Must be kept at reset value
0	RDY	Device ready bit This bit indicates whether the device is ready. The master processor can read this bit to obtain the ready status of the device, after power on, EtherCAT device reset, module reset, digital reset or leave from power savings mode. 0: Device is not ready 1: Device is ready **NOTE**: ■ The rising edge of this bit will set the READYIF bit in INTC_FLAG register, and can trigger an interrupt. ■ When this bit is clear read access to any internal resource is prohibited except for the PMU_CTL, PMU_PDIVAL, and RCU_RSTCFG registers.

- Before this bit is set, write operations to any address are invalid.

2.4.2. Control and status register (PMU_CTL1)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">Reserved</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="15">Reserved</td><td>PMSLPEN</td></tr>
<tr><td colspan="15"></td><td>rw</td></tr>
</table>

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value
0	PMSLPEN	Power management sleep mode enable bit When PMSLPEN is set, the device will enter the power management mode which is configured by PMMODCFG bits. 0: Disable power management sleep mode 1: Enable power management sleep mode **NOTE：** ■ This bit will be cleared when the device wakes up. ■ When PMMODCFG is 0b00 should not set this bit, Although the hardware won't prevent it.

2.4.3. Process data interface reference value register (PMU_PDIREFVAL)

Address offset: 0x1C

Reset value: 0x7654 3210

This register can be accessed by word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">PDIVAL[31:16]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">PDIVAL [15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bits	Fields	Descriptions
31:0	PDIVAL[31:0]	When process data interface(PDI) is ready, reading this register returns the reset value, otherwise, other invalid values are returned (not reset value). Used for PDI

interface testing.

3. Reset and clock unit (RCU)

3.1. Reset control unit (RCTL)

3.1.1. Overview

GDSCN reset control unit includes the control of two kinds of reset: system reset and module reset. System reset includes power-on reset (POR), external pin reset (RSTN) and EtherCAT system reset, which can reset all circuits in the device. Module reset includes digital reset, PHY reset and EtherCAT core reset, which can reset each corresponding module.

3.1.2. Characteristics

- System reset, reset all circuits in the device.
- Multi-module reset, reset the digital circuit except PHY.
- Single-module reset, reset EtherCAT core and external PHY.

3.1.3. Function overview

System reset

System reset can reset the entire device, including power on reset (POR), external pin reset (RSTN) and EtherCAT system level reset, described as follows:

Power-on reset: A power-on reset occurs when the device has just been powered on or when the power is disconnected and reapplied to the device.

RSTN pin reset: Driving the RSTN input pin to low initiates an external pin reset.

EtherCAT system reset: EtherCAT system reset is initiated by a special sequence of three separate consecutive frames/commands.

Module reset

A module reset affects one or more modules and can generate a reset for a variety of modules, as described below:

Multi-module reset: Performs a digital reset by setting the DRST bit of the configuration register (RCU_RSTCFG). A digital reset resets all submodules of the device except the Ethernet PHY.

Single-module reset: A single-module reset resets only the specified module. Single-module reset does not latch configuration pins and includes port A PHY reset, port B PHY reset, and EtherCAT controller reset.

The single module reset is described as follows:

Port A PHY reset is performed by setting the PHYARST bit in the reset configuration register (RCU_RSTCFG) or MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL). After port A PHY is reset, the PHYARST bit and soft reset bit are cleared automatically. The other modules of the device are not affected by this reset. The completion of the PHY reset on port A can be determined by whether the PHYARST bit in the polling reset configuration register (RCU_RSTCFG) or the MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL) is cleared.

Port B PHY reset is performed by setting the PHYBRST bit in the reset configuration register (RCU_RSTCFG) or MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL). After the PHY of port B is reset, the PHYBRST bit and soft reset bit are cleared automatically. The other modules of the device are not affected by this reset. The completion of the PHY reset on port B can be determined by whether the PHYBRST bit in the polling reset configuration register (RCU_RSTCFG) or the MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL) is cleared.

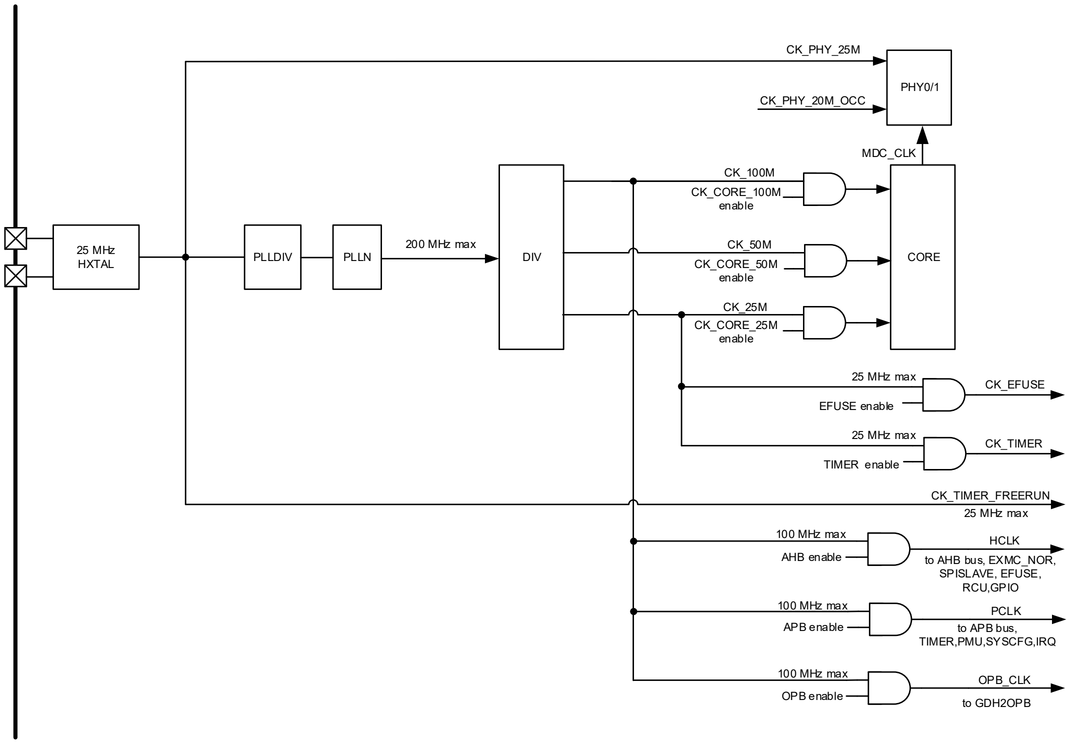
An individual reset of the EtherCAT controller can be performed by resetting ESCRST bit in the configuration register (RCU_RSTCFG). This will reset the EtherCAT core and its registers.

3.2. Clock control unit (CCTL)

3.2.1. Overview

The EtherCAT clock control unit consists primarily of an external High Speed crystal oscillator (HXTAL) and a phase-locked loop (PLL). This clock is usually provided by the OSCIN and OSCOUT of the passive 25MHz crystal oscillator or by the OSCIN pin of the single-ended 25MHz clock source driver.

Figure 3-1. Clock tree



3.2.2. Characteristics

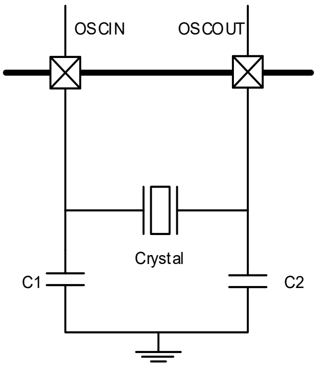
- 25 MHz High speed crystal oscillator (HXTAL).
- A phase locked loop (PLL).

3.2.3. Function overview

High speed crystal oscillator (HXTAL)

The device requires a fixed frequency 25 MHz clock source for use by the internal clock oscillator and PLL. This is usually provided by connecting the 25 MHz crystal oscillator to the OSCIN and OSCOUT pins of the chip. This clock can also be provided by using a single-ended 25 MHz clock source driven OSCIN input pin. If a single-ended source is selected, the clock input must run continuously for the device to function properly. Power-saving mode allows the oscillator or external clock input to pause.

Figure 3-2. HXTAL clock source



The HXTALSTB flag in clock configuration register (RCU_CLKCFG) indicates if the high-speed external crystal oscillator is stable. When the HXTAL is powered up, it will not be released for use until this HXTALSTB bit is set by the hardware. This specific delay period is known as the oscillator “Start-up time”. At this point the HXTAL clock can be used directly as the PLL input clock.

Phase locked loop (PLL)

The PLL input is a 25MHz HXTAL clock, and after PLLN (8) frequency doubling, CK_PLL (200MHz) is obtained. The CK_PLL clock is divided 2 / 4/ 8 by PLLDIV to obtain 100MHz, 50MHz, 25MHz clocks for EtherCAT kernel, and corresponding clocks can be turned on in the core enable register (RCU_COREEN). The clock of the corresponding module can be enabled in the AHB enable register (RCU_AHBEN), APB enable register (RCU_APBEN) and core enable register (RCU_COREEN).

3.3. Register definition

RCU base address: 0x3400

3.3.1. AHB enable register (RCU_AHBEN)

Address offset: 0x00

Reset value: 0x0000 000F

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														GPIOEN	OPBEN
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.

1	GPIOEN	GPIO clock enable This bit is set and reset by software. 0: Disabled GPIO clock 1: Enabled GPIO clock
0	OPBEN	OPB clock enable This bit is set and reset by software. 0: Disabled OPB clock 1: Enabled OPB clock

3.3.2. APB enable register (RCU_APBEN)

Address offset: 0x04

Reset value: 0x0000 0077

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TIMERFUNEN	TIMEREN	PMUEN	Reserved	SYSCFGEN	IRQEN	Reserved
									rw	rw	rw		rw	rw	

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TIMERFUNEN	TIMER function clock enable This bit is set and reset by software. 0: Disabled TIMER function clock 1: Enabled TIMER function clock
5	TIMEREN	TIMER clock enable This bit is set and reset by software. 0: Disabled TIMER clock 1: Enabled TIMER clock (CK_TIMER)
4	PMUEN	PMU clock enable This bit is set and reset by software. 0: Disabled PMU clock 1: Enabled PMU clock
3	Reserved	Must be kept at reset value.
2	SYSCFGEN	SYSCFG clock enable This bit is set and reset by software. 0: Disabled SYSCFG clock

		1: Enabled SYSCFG clock
1	IRQEN	IRQ clock enable This bit is set and reset by software. 0: Disabled IRQ clock 1: Enabled IRQ clock
0	Reserved	Must be kept at reset value.

3.3.3. Core enable register (RCU_COREEN)

Address offset: 0x08

Reset value: 0x0000 0007

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CORE100MEN	CORE50MEN	CORE25MEN
													rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	CORE100MEN	EtherCAT core 100M clock enable This bit is set and reset by software. 0: Disabled CK_CORE_100M clock 1: Enabled CK_CORE_100M clock
1	CORE50MEN	EtherCAT core 50M clock enable This bit is set and reset by software. 0: Disabled CK_CORE_50M clock 1: Enabled CK_CORE_50M clock
0	CORE25MEN	EtherCAT core 25M clock enable This bit is set and reset by software. 0: Disabled CK_CORE_25M clock 1: Enabled CK_CORE_25M clock

3.3.4. Clock configuration register (RCU_CLKCFG)

Address offset: 0x0C

Reset value: 0x0528 0400

Note: PLLBWCTL, PLLN and PLLDIV can only be read and written when PLL_CFG_KEY =

1.

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PLLBWCTL[3:0]				Reserved		PLLN[5:0]					
				rw						rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					PLLDIV[2:0]			Reserved		PLLSTB	Reserved	HXTALSTB	Reserved		
					rw					r		r			

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	PLLBWCTL[3:0]	PLL band width control signal
23:22	Reserved	Must be kept at reset value.
21:16	PLLN[5:0]	PLL clock multiplication factor This bit is set and reset by software. 000000: Reserved 000001: Reserved … 000110: Reserved 000111: Reserved 001000: Multiplication factor is 8 001001: Multiplication factor is 9 001010: Multiplication factor is 10 … 111110: Multiplication factor is 62 111111: Multiplication factor is 63
15:11	Reserved	Must be kept at reset value.
10:8	PLLDIV[2:0]	PLL clock frequency division factor This bit is set and reset by software. 000: Division factor is 1 001: Division factor is 2 … 110: Division factor is 7 111: Division factor is 8
7:6	Reserved	Must be kept at reset value.
5	PLLSTB	PLL clock stabilization flag Set by hardware to indicate if the PLL output clock is stable and ready for use. 0: PLL is not stable

		1: PLL is stable
4	Reserved	Must be kept at reset value.
3	HXTALSTB	High speed crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
2:0	Reserved	Must be kept at reset value.

3.3.5. Reset configuration register (RCU_RSTCFG)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						PHYBREL	PHYAREL	Reserved	ESCRST	Reserved			PHYBRST	PHYARST	DRST
						rw	rw		rw				rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	PHYBREL	Port B PHY release 0: Port B PHY remains in the reset state 1: Port B PHY releases from reset **Note**: This bit is valid when phyrst_mode = 1
8	PHYAREL	Port A PHY release 0: Port A PHY remains in the reset state 1: Port A PHY releases from reset **Note**: This bit is valid when phyrst_mode = 1
7	Reserved	Must be kept at reset value.
6	ESCRST	EtherCAT reset Setting this bit to 1 will reset the EtherCAT core. When the EtherCAT core is released from the reset state, this bit is automatically cleared by the hardware. When this bit is set, all writes to this bit are ignored.
5:3	Reserved	Must be kept at reset value.
2	PHYBRST	Port B PHY reset

		Setting this bit to 1 will reset the port B PHY. When port B PHY is released from the reset state, this bit is automatically cleared by the hardware. When the bit is set, all writes to that bit are ignored.
1	PHYARST	Port A PHY reset Setting this bit to 1 will reset the port A PHY. When port A PHY is released from the reset state, this bit is automatically cleared by the hardware. When the bit is set, all writes to that bit are ignored.
0	DRST	Digital reset Setting this bit to 1 will reset the entire chip (except for the PLL, port B PHY and port A PHY). When the chip is released from the reset state, this bit is automatically cleared by the hardware. When this bit is set, all writes to this bit are ignored.

3.3.6. PLL configuration key register (RCU_PLL_CFG_KEY)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PLL_CFG _KEY
															rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	PLL_CFG_KEY	When the register is written to 0x78b465a1, the bit is 1 and PLLBWCTL, PLLN and PLLDIV bits in RCU_CLKCFG register can be read and written.

3.3.7. Pin reset flag register (RCU_PRSTF)

Address offset: 0x18

Reset value: 0x0000 0002

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PRSTF	PRSTC
														r	rw

Bits	Fields	Descriptions

31:2	Reserved	Must be kept at reset value.
1	PRSTF	Pin reset flag This bit is set when pin reset occurs and cleared when PRSTC is set. 0: No pin reset occurs 1: Pin reset occurs
0	PRSTC	Pin reset flag clear Write 1 by software to reset the PRSTF flag

4. Interrupt controller (INTC)

4.1. Overview

The multi-layer interrupt structure of the device is programmable and controlled by the interrupt controller (INTC). Interrupt events are generated internally by individual submodules and can be configured to output a single external host interrupt via the IRQ pin.

4.2. Characteristics

- The IRQ interrupt buffer mode, polarity, and de-assertion interval can be modified.
- The IRQ interrupt can be set the output mode to open-drain, enabling multiple devices to share the interrupt.
- All internal interrupts can be masked and trigger the IRQ interrupt.
- The device supports the following 8 types interrupts:
  - Software interrupt.
  - Device ready interrupt.
  - Ethernet PHY interrupt.
  - Timer interrupt.
  - PME interrupt.
  - AHB2OPB bridge interrupt.
  - EtherCAT interrupt.
  - Clock output test mode.

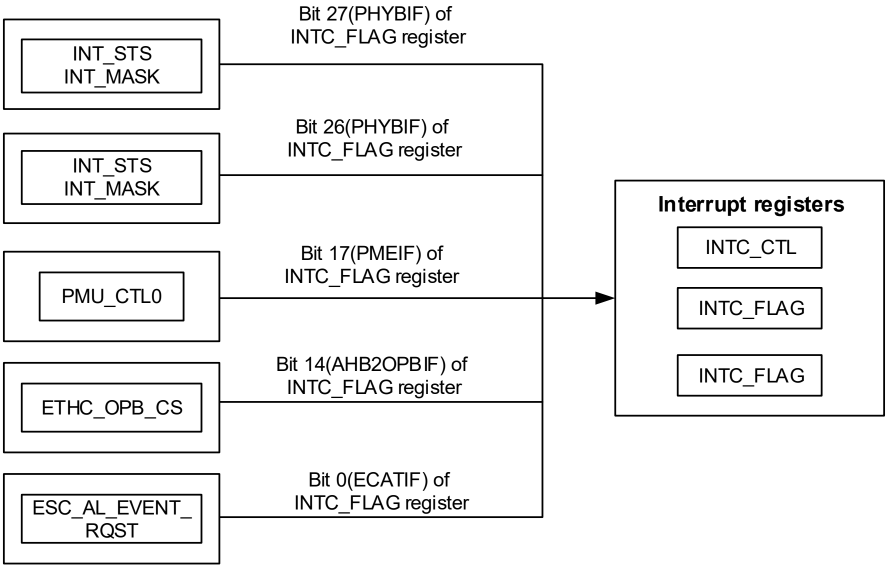
4.3. Interrupts function overview

The interrupt of the device can be divided into the following two types according to whether the interrupt source is enabled and cleared in the register of the submodule:

- The first type includes software, device ready, and timer interrupts that are directly accessed and configured (including monitored, enabled / disabled, and cleared) through INTC_FLAG register and INTC_EN register.
- The second type includes Ethernet PHY, power management, AHB2OPB bridge, and EtherCAT interrupts. INTC_FLAG register can provide indications of these interrupt events, but has no specific information of interrupt source. Software needs to poll an submodule interrupt register of the to determine the interrupt source. INTC_FLAG register can be cleared only after the interrupt has been processed and the interrupt source cleared.

Interrupt events can trigger external IRQ interrupt pin output. By configuring INTC_CTL register, user can enable / disable IRQ interrupt pin output and configure IRQ interrupt buffer mode, polarity and de-assertion interval. DEAS field of INTC_CTL register is used to configure the interrupt request de-assertion interval, which guarantees that the minimum IRQ interrupt output de-assertion interval period, and that de-assertion interval always starts when IRQ pin is set to de-assertion. The relationship between interrupt register and interrupt source control register, as shown in ***Figure 4-1. Block diagram of interrupt***.

Figure 4-1. Block diagram of interrupt


INT_STS
INT_MASK
Bit 27(PHYBIF) of INTC_FLAG register
INT_STS
INT_MASK
Bit 26(PHYBIF) of INTC_FLAG register
PMU_CTL0
Bit 17(PMEIF) of INTC_FLAG register
ETHC_OPB_CS
Bit 14(AHB2OPBIF) of INTC_FLAG register
ESC_AL_EVENT_RQST
Bit 0(ECATIF) of INTC_FLAG register
Interrupt registers
INTC_CTL
INTC_FLAG
INTC_FLAG

4.3.1. Software interrupt

Interrupt controller provides control over a general purpose software interrupt. When SWIE bit of INTC_EN register is switched from 0 to 1, SWIF bit of INTC_FLAG register is set. This interrupt provides a relatively simple method of generating interrupts in software and is used in conventional software design.

In order for a software interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.2. Device ready interrupt

Interrupt controller provides control over a device ready interrupt. When READYIE bit of INTC_EN register is switched from 0 to 1, READYIF bit of INTC_FLAG register is used to indicate that the device is ready for access after power-on or reset condition.

In order for a device ready interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.3. Ethernet PHY interrupt

Interrupt controller provides control over a ethernet PHY interrupt. When PHYAIE bit of INTC_EN register is switched from 0 to 1, PHYAIF and PHYBIF bits of INTC_FLAG register are used to indicate interrupt events from the Ethernet PHY. For more information about Ethernet PHY interrupt sources, refer to the ***Ethernet PHYS***.

In order for a ethernet PHY interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.4. Timer interrupt

Interrupt controller provides control over a timer interrupt. This interrupt is generated when the value of timer count register changes from 0 to 0xFFFF. When TIMIE bit of INTC_EN register is switched from 0 to 1, TIMIF bit of INTC_FLAG register is used to indicate interrupt events from the timer.

In order for a timer interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.5. PME interrupt

Interrupt controller provides control over a PME interrupt. When PMEIE bit of INTC_EN register is switched from 0 to 1, PMEIF bit of INTC_FLAG register is used to indicate interrupt events from the PMU. For more information about PMU interrupt sources, refer to the ***Power management unit (PMU)***.

In order for a power management interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.6. AHB2OPB bridge interrupt

Interrupt controller provides control over a AHB2OPB bridge interrupt. When AHB2OPBIE bit of INTC_EN register is switched from 0 to 1, AHB2OPBIF bit of INTC_FLAG register is used to indicate the AHB2OPB bridge interrupt event from the BUS. For more information about SYS interrupt sources, refer to the ***System and bus architecture***.

In order for a AHB2OPB bridge interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.7. EtherCAT interrupt

Interrupt controller provides control over a EtherCAT interrupt. When ECATIE bit of INTC_EN register is switched from 0 to 1, ECATIF bit of INTC_FLAG register is used to indicate interrupt events from the EtherCAT. For more information about EtherCAT interrupt sources, refer to the ***EtherCAT***.

In order for a EtherCAT interrupt event to trigger the external IRQ interrupt pin, IRQ output must be enabled by setting IRQEN bit of INTC_CTL register to 1.

4.3.8. Clock output test mode

In order to debug system and observe the clock, the IRQ pin output crystal oscillator clock can be realized by setting IRQCKOUT bit of INTC_CTL register to 1. At this point, the IRQ pin must be configured in push-pull output mode (IRQMODE=1) for best results.

4.4. Register definition

INTC base address: 0x3A00

4.4.1. Control register (INTC_CTL)

Address offset: 0x00

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEAS[7:0]								Reserved							
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	DEASC	DEASSTAT	IRQSTAT	Reserved			IRQEN	Reserved			IRQPOL	Reserved		IRQCKOUT	IRQMODE
	rw	r	r				rw				rw			rw	rw

Bits	Fields	Descriptions
31:24	DEAS[7:0]	Interrupt de-assertion interval These bits are used to configure the de-assertion interval, uints in 10us. When setting these bits to 0, which disables the DEAS interval, reset the interval counter, and send any pending interrupts. If these bits are set to a non-zero value, any subsequent interrupts will follow this setting.
23:15	Reserved	Must be kept at reset value.
14	DEASC	Interrupt de-assertion interval clear When setting this bit is to 1, the de-assertion interval counter will be cleared and a new de-assertion interval count will be enabled (regardless of whether the de-assertion interval is currently active or not). This bit is automatically cleared by hardware. This bit is not reset when a software reset occurs on the device. 0: No effect 1: Clear the de-assertion interval counter
13	DEASSTAT	Interrupt de-assertion interval status This bit is used to indicate the status of the interrupt de-assertion interval. 0: Not in de-assertion interval (interrupts will be sent to the IRQ pin) 1: In de-assertion interval (interrupts will not be sent to the IRQ pin)
12	IRQSTAT	Internal IRQ line status This bit is used to indicate the status of the internal IRQ line and is not affected by the setting of the IRQEN bit. 0: None of the enabled interrupts active 1: At least one enabled interrupt active

11:9	Reserved	Must be kept at reset value.
8	IRQEN	IRQ pin output enable This bit controls the interrupt output function of the IRQ pin. 0: Disable IRQ pin output 1: Enable IRQ pin output
7:5	Reserved	Must be kept at reset value.
4	IRQPOL	IRQ pin output polarity This bit is not reset when a digital reset occurs on the device. This bit is reset when a system reset occurs on the device, such as power-on reset, pin reset and EtherCAT system reset. 0: IRQ pin output activation level is low 1: IRQ pin output activation level is high **Note:** When IRQ pin is set to output open-drain mode (IRQMODE=0), this bit is ignored and the IRQ pin output activation level is always low.
3:2	Reserved	Must be kept at reset value.
1	IRQCKOUT	IRQ clock output 0: No clock output 1: IRQ pin output crystal oscillator clock (Used for system debugging, observing the clock) **Note:** When this bit is set to 1, the IRQ pin must be set to push-pull output mode (IRQMODE=1).
0	IRQMODE	IRQ pin output mode This bit is not reset when a digital reset occurs on the device.This bit is reset when a system reset occurs on the device, such as power-on reset, pin reset and EtherCAT system reset. 0: Output open-drain mode 1: Output push-pull mode

4.4.2. Flag register (INTC_FLAG)

Address offset: 0x04

Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIF	READYIF	Reserved		PHYBIF	PHYAIF	Reserved						TIMIF	Reserved	PMEIF	Reserved
rc_w1	rc_w1			r	r							rc_w1		rc_w1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	AHB2OP BIF	Reserved													ECATIF
	r														r

Bits	Fields	Descriptions
31	SWIF	Software interrupt flag This interrupt is generated when SWIE bit in INTC_EN register is 1. The software can clear it by writing 1.
30	READYIF	Device ready interrupt flag This interrupt is used to indicate that the device is ready for access after power-on or reset condition. The software can clear it by writing 1.
29:28	Reserved	Must be kept at reset value.
27	PHYBIF	Ethernet PHY B interrupt flag This bit indicates an Ethernet PHY B interrupt event.
26	PHYAIF	Ethernet PHY A interrupt flag This bit indicates an Ethernet PHY A interrupt event.
25:20	Reserved	Must be kept at reset value.
19	TIMIF	Timer interrupt flag This interrupt is generated when the timer counter register changes from 0 to 0xFFFF. The software can clear it by writing 1.
18	Reserved	Must be kept at reset value.
17	PMEIF	PME interrupt flag This interrupt is generated when the PMU detects a power management event configured in the PMU_CTL register. The software can clear it by writing 1. **Note:** Interrupt de-assertion interval does not apply to PMU interrupt.
16:15	Reserved	Must be kept at reset value.
14	AHB2OPBIF	AHB2OPB bridge interrupt flag This bit indicates an AHB2OPB bridge interrupt event from the BUS.
13:1	Reserved	Must be kept at reset value.
0	ECATIF	EtherCAT interrupt flag This bit indicates an interrupt event from EtherCAT.

4.4.3. Enable register (INTC_EN)

Address offset: 0x08
Reset value: 0x0000 0000

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIE	READYIE	Reserved		PHYBIE	PHYAIE	Reserved						TIMIE	Reserved	PMEIE	Reserved
rw	rw			rw	rw							rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	AHB2OP BIE	Reserved													ECATIE
	rw														rw

Bits	Fields	Descriptions
31	SWIE	Software interrupt enable 0: Disable software interrupt 1: Enable software interrupt
30	READYIE	Device ready interrupt enable 0: Disable device ready interrupt 1: Enable device ready interrupt
29:28	Reserved	Must be kept at reset value.
27	PHYBIE	Ethernet PHY B interrupt enable 0: Disable ethernet PHY B interrupt 1: Enable ethernet PHY B interrupt
26	PHYAIE	Ethernet PHY A interrupt enable 0: Disable ethernet PHY A interrupt 1: Enable ethernet PHY A interrupt
25:20	Reserved	Must be kept at reset value.
19	TIMIE	Timer interrupt enable 0: Disable timer interrupt 1: Enable timer interrupt
18	Reserved	Must be kept at reset value.
17	PMEIE	PME interrupt enable 0: Disable power management interrupt 1: Enable power management interrupt
16:15	Reserved	Must be kept at reset value.
14	AHB2OPBIE	AHB2OPB bridge interrupt enable 0: Disable AHB2OPB bridge interrupt 1: Enable AHB2OPB bridge interrupt
13:1	Reserved	Must be kept at reset value.
0	ECATIE	EtherCAT interrupt enable 0: Disable EtherCAT interrupt

1: Enable EtherCAT interrupt

5. General-purpose I/Os (GPIO)

5.1. Overview

There are up to 35 general purpose I/O pins (GPIO), Each GPIO port will determine the current capabilities of that port based on the current operating mode of the chip, including input/output modes. Each of the GPIO pins can be configured as a pull-up/pull-down or floating. When the pin is in output mode, the pin can be configured as a push-pull/drain open/source open drain output.

5.2. Characteristics

- Each pin weak pull-up/pull-down function.
- Output push-pull/open-drain enable control.
- Configures the function of the selected pin according to the chip mode.

5.3. Function overview

5.3.1. GPIO pin configuration

When is the reset, All the GPIO ports are configured as the input floating mode that input disabled without pull-up(PU) / pull-down(PD) resistors. After the chip is reset, wait for the EEPROM to load. When the loading is complete, determine the initial state of the pin after the chip is reset according to ESC PDI_TYPE.When PDI_TYPE equal to 0x04 select Digital IO mode, When PDI_TYPE equal to 0x80 select SPI mode or EXMC mode.

The GPIO pin is controlled as input or output state according to the working state.
All GPIO pins have an internal weak pull-up and weak pull-down option When the GPIO pin is configured as an output pin, it can configure the output drive mode: push-pull or drain open drain and source open drain mode. The pull-down mode and output mode configurations support writing via EXMC or SPI communication.

5.3.2. External interrupt / event lines

Only one external interrupt output interface is supported. The interrupt output configuration is determined by the internal register of the chapter ***Reset and clock unit*** (RCU)

5.4. Reset control unit (RCTL)

5.4.1. Overview

GDSCN reset control unit includes the control of two kinds of reset: system reset and module reset. System reset includes power-on reset (POR), external pin reset (RSTN) and EtherCAT system reset, which can reset all circuits in the device. Module reset includes digital reset, PHY reset and EtherCAT core reset, which can reset each corresponding module.

5.4.2. Characteristics

- System reset, reset all circuits in the device.
- Multi-module reset, reset the digital circuit except PHY.
- Single-module reset, reset EtherCAT core and external PHY.

5.4.3. Function overview

System reset

System reset can reset the entire device, including power on reset (POR), external pin reset (RSTN) and EtherCAT system level reset, described as follows:

Power-on reset: A power-on reset occurs when the device has just been powered on or when the power is disconnected and reapplied to the device.

RSTN pin reset: Driving the RSTN input pin to low initiates an external pin reset.

EtherCAT system reset: EtherCAT system reset is initiated by a special sequence of three separate consecutive frames/commands.

Module reset

A module reset affects one or more modules and can generate a reset for a variety of modules, as described below:

Multi-module reset: Performs a digital reset by setting the DRST bit of the configuration register (RCU_RSTCFG). A digital reset resets all submodules of the device except the Ethernet PHY.

Single-module reset: A single-module reset resets only the specified module. Single-module reset does not latch configuration pins and includes port A PHY reset, port B PHY reset, and EtherCAT controller reset.

The single module reset is described as follows:

Port A PHY reset is performed by setting the PHYARST bit in the reset configuration register (RCU_RSTCFG) or MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL). After port A PHY is reset, the PHYARST bit and soft reset bit are cleared automatically. The other modules of the device are not affected by this reset. The completion of the PHY reset on port A can be determined by whether the PHYARST bit in the polling reset configuration register (RCU_RSTCFG) or the MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL) is cleared.

Port B PHY reset is performed by setting the PHYBRST bit in the reset configuration register (RCU_RSTCFG) or MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL). After the PHY of port B is reset, the PHYBRST bit and soft reset bit are cleared automatically. The other modules of the device are not affected by this reset. The completion of the PHY reset on port B can be determined by whether the PHYBRST bit in the polling reset configuration register (RCU_RSTCFG) or the MR_MAIN_REST bit in the PHY control register (PHY_MII_CTL) is cleared.

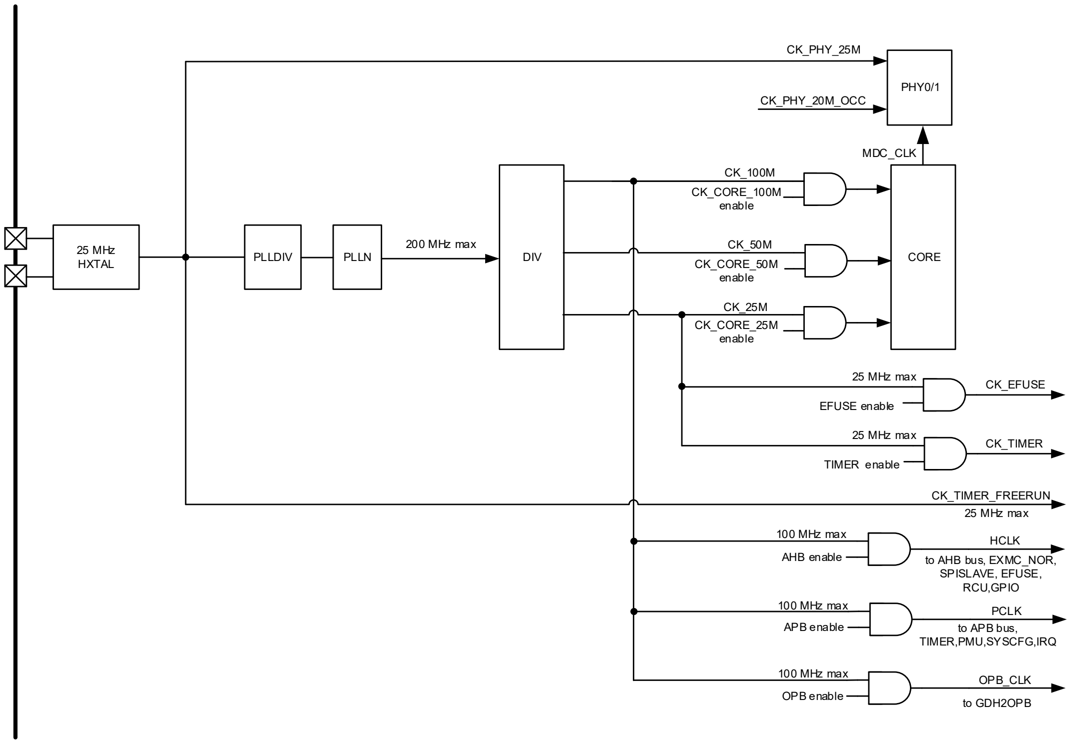
An individual reset of the EtherCAT controller can be performed by resetting ESCRST bit in the configuration register (RCU_RSTCFG). This will reset the EtherCAT core and its registers.

5.5. Clock control unit (CCTL)

5.5.1. Overview

The EtherCAT clock control unit consists primarily of an external High Speed crystal oscillator (HXTAL) and a phase-locked loop (PLL). This clock is usually provided by the OSCIN and OSCOUT of the passive 25MHz crystal oscillator or by the OSCIN pin of the single-ended 25MHz clock source driver.

Figure 3-1. Clock tree



5.5.2. Characteristics

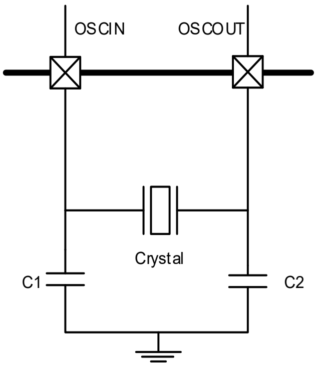
- 25 MHz High speed crystal oscillator (HXTAL).
- A phase locked loop (PLL).

5.5.3. Function overview

High speed crystal oscillator (HXTAL)

The device requires a fixed frequency 25 MHz clock source for use by the internal clock oscillator and PLL. This is usually provided by connecting the 25 MHz crystal oscillator to the OSCIN and OSCOUT pins of the chip. This clock can also be provided by using a single-ended 25 MHz clock source driven OSCIN input pin. If a single-ended source is selected, the clock input must run continuously for the device to function properly. Power-saving mode allows the oscillator or external clock input to pause.

Figure 3-2. HXTAL clock source



The HXTALSTB flag in clock configuration register (RCU_CLKCFG) indicates if the high-speed external crystal oscillator is stable. When the HXTAL is powered up, it will not be released for use until this HXTALSTB bit is set by the hardware. This specific delay period is known as the oscillator “Start-up time”. At this point the HXTAL clock can be used directly as the PLL input clock.

Phase locked loop (PLL)

The PLL input is a 25MHz HXTAL clock, and after PLLN (8) frequency doubling, CK_PLL (200MHz) is obtained. The CK_PLL clock is divided 2 / 4/ 8 by PLLDIV to obtain 100MHz, 50MHz, 25MHz clocks for EtherCAT kernel, and corresponding clocks can be turned on in the core enable register (RCU_COREEN). The clock of the corresponding module can be enabled in the AHB enable register (RCU_AHBEN), APB enable register (RCU_APBEN) and core enable register (RCU_COREEN).

5.6. Register definition

RCU base address: 0x3400

5.6.1. AHB enable register (RCU_AHBEN)

Address offset: 0x00
Reset value: 0x0000 000F

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														GPIOEN	OPBEN
														rw	rw

Bits	Fields	Descriptions
31:2	Reserved	Must be kept at reset value.

1	GPIOEN	GPIO clock enable This bit is set and reset by software. 0: Disabled GPIO clock 1: Enabled GPIO clock
0	OPBEN	OPB clock enable This bit is set and reset by software. 0: Disabled OPB clock 1: Enabled OPB clock

5.6.2. APB enable register (RCU_APBEN)

Address offset: 0x04

Reset value: 0x0000 0077

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TIMERFUNEN	TIMEREN	PMUEN	Reserved	SYSCFGEN	IRQEN	Reserved
									rw	rw	rw		rw	rw	

Bits	Fields	Descriptions
31:7	Reserved	Must be kept at reset value.
6	TIMERFUNEN	TIMER function clock enable This bit is set and reset by software. 0: Disabled TIMER function clock 1: Enabled TIMER function clock
5	TIMEREN	TIMER clock enable This bit is set and reset by software. 0: Disabled TIMER clock 1: Enabled TIMER clock (CK_TIMER)
4	PMUEN	PMU clock enable This bit is set and reset by software. 0: Disabled PMU clock 1: Enabled PMU clock
3	Reserved	Must be kept at reset value.
2	SYSCFGEN	SYSCFG clock enable This bit is set and reset by software. 0: Disabled SYSCFG clock

		1: Enabled SYSCFG clock
1	IRQEN	IRQ clock enable This bit is set and reset by software. 0: Disabled IRQ clock 1: Enabled IRQ clock
0	Reserved	Must be kept at reset value.

5.6.3. Core enable register (RCU_COREEN)

Address offset: 0x08

Reset value: 0x0000 0007

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													CORE100MEN	CORE50MEN	CORE25MEN
													rw	rw	rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value.
2	CORE100MEN	EtherCAT core 100M clock enable This bit is set and reset by software. 0: Disabled CK_CORE_100M clock 1: Enabled CK_CORE_100M clock
1	CORE50MEN	EtherCAT core 50M clock enable This bit is set and reset by software. 0: Disabled CK_CORE_50M clock 1: Enabled CK_CORE_50M clock
0	CORE25MEN	EtherCAT core 25M clock enable This bit is set and reset by software. 0: Disabled CK_CORE_25M clock 1: Enabled CK_CORE_25M clock

5.6.4. Clock configuration register (RCU_CLKCFG)

Address offset: 0x0C

Reset value: 0x0528 0400

Note: PLLBWCTL, PLLN and PLLDIV can only be read and written when PLL_CFG_KEY =

1.

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PLLBWCTL[3:0]				Reserved		PLLN[5:0]					
				rw						rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					PLLDIV[2:0]			Reserved		PLLSTB	Reserved	HXTALSTB	Reserved		
					rw					r		r			

Bits	Fields	Descriptions
31:28	Reserved	Must be kept at reset value.
27:24	PLLBWCTL[3:0]	PLL band width control signal
23:22	Reserved	Must be kept at reset value.
21:16	PLLN[5:0]	PLL clock multiplication factor This bit is set and reset by software. 000000: Reserved 000001: Reserved … 000110: Reserved 000111: Reserved 001000: Multiplication factor is 8 001001: Multiplication factor is 9 001010: Multiplication factor is 10 … 111110: Multiplication factor is 62 111111: Multiplication factor is 63
15:11	Reserved	Must be kept at reset value.
10:8	PLLDIV[2:0]	PLL clock frequency division factor This bit is set and reset by software. 000: Division factor is 1 001: Division factor is 2 … 110: Division factor is 7 111: Division factor is 8
7:6	Reserved	Must be kept at reset value.
5	PLLSTB	PLL clock stabilization flag Set by hardware to indicate if the PLL output clock is stable and ready for use. 0: PLL is not stable

		1: PLL is stable
4	Reserved	Must be kept at reset value.
3	HXTALSTB	High speed crystal oscillator (HXTAL) clock stabilization flag Set by hardware to indicate if the HXTAL oscillator is stable and ready for use. 0: HXTAL oscillator is not stable 1: HXTAL oscillator is stable
2:0	Reserved	Must be kept at reset value.

5.6.5. Reset configuration register (RCU_RSTCFG)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						PHYBREL	PHYAREL	Reserved	ESCRST	Reserved			PHYBRST	PHYARST	DRST
						rw	rw		rw				rw	rw	rw

Bits	Fields	Descriptions
31:10	Reserved	Must be kept at reset value.
9	PHYBREL	Port B PHY release 0: Port B PHY remains in the reset state 1: Port B PHY releases from reset **Note**: This bit is valid when phyrst_mode = 1
8	PHYAREL	Port A PHY release 0: Port A PHY remains in the reset state 1: Port A PHY releases from reset **Note**: This bit is valid when phyrst_mode = 1
7	Reserved	Must be kept at reset value.
6	ESCRST	EtherCAT reset Setting this bit to 1 will reset the EtherCAT core. When the EtherCAT core is released from the reset state, this bit is automatically cleared by the hardware. When this bit is set, all writes to this bit are ignored.
5:3	Reserved	Must be kept at reset value.
2	PHYBRST	Port B PHY reset

		Setting this bit to 1 will reset the port B PHY. When port B PHY is released from the reset state, this bit is automatically cleared by the hardware. When the bit is set, all writes to that bit are ignored.
1	PHYARST	Port A PHY reset Setting this bit to 1 will reset the port A PHY. When port A PHY is released from the reset state, this bit is automatically cleared by the hardware. When the bit is set, all writes to that bit are ignored.
0	DRST	Digital reset Setting this bit to 1 will reset the entire chip (except for the PLL, port B PHY and port A PHY). When the chip is released from the reset state, this bit is automatically cleared by the hardware. When this bit is set, all writes to this bit are ignored.

5.6.6. PLL configuration key register (RCU_PLL_CFG_KEY)

Address offset: 0x14
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															PLL_CFG_KEY
															rw

Bits	Fields	Descriptions
31:1	Reserved	Must be kept at reset value.
0	PLL_CFG_KEY	When the register is written to 0x78b465a1, the bit is 1 and PLLBWCTL, PLLN and PLLDIV bits in RCU_CLKCFG register can be read and written.

5.6.7. Pin reset flag register (RCU_PRSTF)

Address offset: 0x18
Reset value: 0x0000 0002

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														PRSTF	PRSTC
														r	rw

Bits	Fields	Descriptions

31:2	Reserved	Must be kept at reset value.
1	PRSTF	Pin reset flag This bit is set when pin reset occurs and cleared when PRSTC is set. 0: No pin reset occurs 1: Pin reset occurs
0	PRSTC	Pin reset flag clear Write 1 by software to reset the PRSTF flag

Interrupt controller (INTC), and the output mode is also determined by the internal bit.

5.6.8. Alternate functions (AF)

When the chip is in different modes, each pin has different functions.

Digital IO mode: When PDI_TYPE = 0x04, AFIO is adjusted to digital IO mode.

EXMC mode: When PDI_TYPE = 0x80 and the pad of MCU_PDI_TYPE = 1. AFIO is adjusted to EXMC mode.

SPI mode: When PDI_TYPE equal to 0x80 and the pad of MCU_PDI_TYPE equal to 0. AFIO is adjusted to SPI mode.

Table 5-1. GPIO configuration table

Mode name	Register/signal	Description

EXMC	pdi_type / mcu_pdi_type / sip_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b1;
		sip_mode == 1'b0

DIO	pdi_type / sip_mode	pdi_type == 0x04; sip_mode == 1'b0

spi8w_gpio	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b0;
		sip_mode == 1'b0;
		line_mode == 2'b11;
		LINKACT_LED1 Pin must be externally drop-down

spi4w_mii_down	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode / chip_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b0;
		sip_mode == 1'b0;
		line_mode == 2'b10;
		chip_mode == 2'b10
		LINKACT_LED1 Pin must be externally pull-up

spi4w_mii_up	pdi_type / mcu_pdi_type / spi_ext_mode / sip_mode / line_mode / chip_mode	pdi_type == 0x80;
		mcu_pdi_type == 1'b0;
		sip_mode == 1'b0;
		line_mode == 2'b10;
		chip_mode == 2'b11;
		LINKACT_LED1 Pin must be externally pull-up

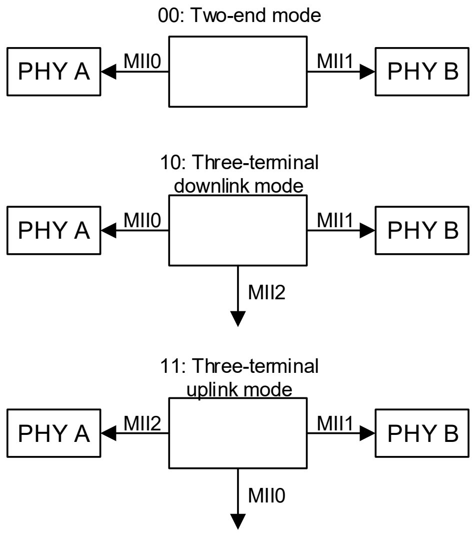
In addition, when in SPI (2 / 4 / 8 wire) +MII and the chip_mode [1:0] is not equal to 0x11, the EtherCAT port 0 is connected to the internal PHY A.

When chip_mode [1:0] is equal to 0x11, the EtherCAT port 0 is connected to the MII pin, port 2 is connected to the internal PHY A.

When chip_mode [1:0] is equal to 0x10b, the EtherCAT port 2 is connected to the MII pin.

When chip_mode [1:0] is equal to 0x00, In this case, the output of SPI+GPIO mode is not affected, and the MII signal is not output in SPI+MII mode.

Figure 5-1. Port line PHYS



Note:

1. Some pins are locked during power-on reset or when RST# is set to invalid, and automatically switch after being locked.
2. The MII_LINKPOL signal is latched after reset to determine the polarity of the MII_LINK pin. If MII_LINKPOL equals 0, it indicates that the MII_LINK pin is active low, signifying that a 100 Mbps full-duplex link has been established. If MII_LINKPOL equals 1, it indicates that the MII_LINK pin is active high, signifying that a 100 Mbps full-duplex link has been established.
3. SYNC1_LATCH1/SYNC0_LATCH0 pad omode / io_en is determined by the ESC internal register.
4. The following latch signals must to be pull up or down in the following mode, and cannot be set to the X state.

   (1). If spi_ext_mode is equal to 1 and inphy_bypass is equal to 0, the chip_mode[1:0] must be set drop-down state.

   (2). The pad of IO16 must be configure as the pull-up or drop-down state

   (3). The pad of EESIZE must be configure as the pull-up or drop-down state

   (4). The pad of IO17 must be configure as the drop-down and drop-down state

   (5). When inphy_bypass is equal to 1, the pad of MII_LINKPOL must be set the pull-up or drop-down state.

PDI_TYPE: Reference to EtherCAT register ***ESC PDI Control register (ESC_PDI_CTL)***

line_mode: The SPI output to GPIO is determined by the SPI input instruction

spi_ext_mode: Reference to System configuration register in ***System configuration register 0 (SYSCFG_CFG0)*** Bit 2.

chip_mode[1:0]: pad of LINKACTLED1/ LINKACTLED0 latched after reset.

inphy_bypass: The register bit configured by factory set to 0, cannot be modified.

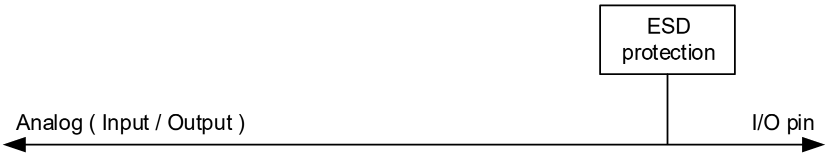
5.6.9. Analog configuration

When GPIO pin is used as analog configuration:

- The weak pull-up and pull-down resistors are disabled.
- The output buffer is disabled.
- The schmitt trigger input is de-activated.
- Read access to the port input status register gets the value “0”.

Figure 5-2. Basic structure of Analog configuration shows the analog configuration of the GPIO pin.

Figure 5-2. Basic structure of Analog configuration



5.6.10. Alternate function (AF) configuration

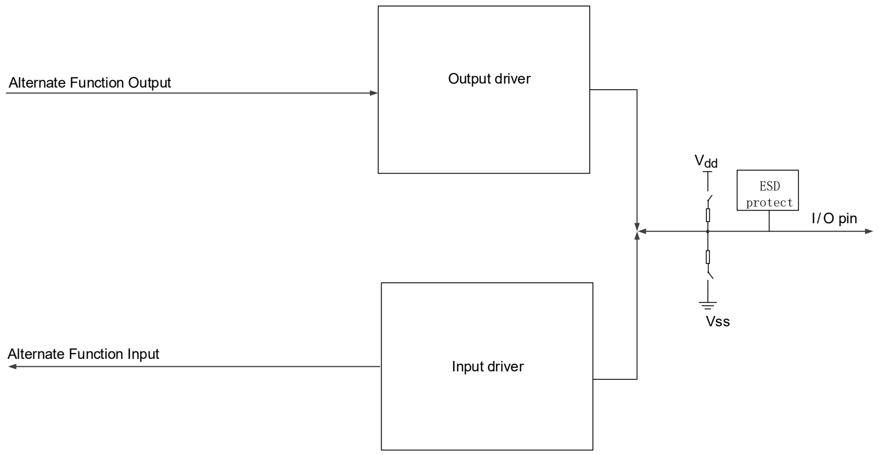
To suit for different device packages, the GPIO supports some alternate functions mapped to some other pins by software.

When be configured as alternate function:

- The output buffer is enabled in open-drain or push-pull configuration.
- The output buffer is driven by the peripheral.
- The schmitt trigger input is activated.
- The weak pull-up and pull-down resistors could be chosen.

Figure 5-3. Basic structure of Alternate function configuration shows the alternate function configuration of the GPIO pin.

Figure 5-3. Basic structure of Alternate function configuration



Note:

In OSPI mode, pdi_gpio15 cannot be used in OSPI+GPIO mode due to the large number of SPI pins occupied. MII_CLK25 cannot be used in OSPI+MII mode.

5.7. Register definition

GPIO base address: 0x3500

5.7.1. Port output mode register0 (GPIO0_OMODE0)

Address offset: 0x00

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OM015[1:0]		OM014[1:0]		OM013[1:0]		OM012[1:0]		OM011[1:0]		OM010[1:0]		Reserved		OM008[1:0]	
rw		rw		rw		rw		rw		rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM007[1:0]		OM006[1:0]		OM005[1:0]		OM004[1:0]		Reserved		OM002[1:0]		OM001[1:0]		OM000[1:0]	
rw		rw		rw		rw				rw		rw		rw	

Bits	Fields	Descriptions
31:30	OM015[1:0]	Pin IO11 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
29:28	OM014[1:0]	Pin OE_EXT output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
27:26	OM013[1:0]	Pin IO4 output mode bit

		These bits are set and cleared by software. Refer to OM000[1:0] description
25:24	OM012[1:0]	Pin IO5 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
23:22	OM011[1:0]	Pin IO6 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
21:20	OM010[1:0]	Pin LATCH_IN output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
19:18	Reserved	Must be kept at reset value.
17:16	OM008[1:0]	Pin WD_STATE output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
15:14	OM007[1:0]	Pin IO7 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
13:12	OM006[1:0]	Pin IO8 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
11:10	OM005[1:0]	Pin EOF output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
9:8	OM004[1:0]	Pin SOF output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
7:6	Reserved	Must be kept at reset value
5:4	OM002[1:0]	Pin IO18 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
3:2	OM001[1:0]	Pin IO17 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
1:0	OM000[1:0]	Pin IO16 output mode bit These bits are set and cleared by software. 00: Output push-pull mode (reset value) 01: Output open-drain mode

10: Output open-source mod

11: Reserved

5.7.2. Port output mode register1 (GPIO0_OMODE1)

Address offset: 0x04

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												OM017[1:0]		OM016[1:0]	
												rw		rw	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value
3:2	OM017[1:0]	Pin IO13 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
1:0	OM016[1:0]	Pin IO12 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description

5.7.3. Port output mode register2 (GPIO1_OMOD0)

Address offset: 0x08

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OM115		OM114		OM113		Reserved		OM111		OM110		Reserved		OM108	
rw		rw		rw				rw		rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OM107		OM106		OM105		Reserved		OM103		OM102		OM101		OM100	
rw		rw		rw				rw		rw		rw		rw	

Bits	Fields	Descriptions
31:30	OM115[1:0]	Pin LINKACTLED0 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description

29:28	OM114[1:0]	Pin LINKACTLED1 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
27:26	OM113[1:0]	Pin EESIZE output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
25:24	Reserved	Must be kept at reset value.
23:22	OM111[1:0]	Pin EESCL output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
21:20	OM110[1:0]	Pin EESDA output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
19:18	Reserved	Must be kept at reset value.
17:16	OM108[1:0]	Pin IO2 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
15:14	OM107[1:0]	Pin IO1 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
13:12	OM106[1:0]	Pin IO0 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
11:10	OM105[1:0]	Pin WD_TRIG output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
9:8	Reserved	Must be kept at reset value.
7:6	OM103[1:0]	Pin IO9 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
5:4	OM102[1:0]	Pin IO15 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
3:2	OM101[1:0]	Pin IO14 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
1:0	OM100[1:0]	Pin IO10 output mode bit

These bits are set and cleared by software.

Refer to OM000[1:0] description

5.7.4. Port output mode register3 (GPIO1_OMOD1)

Address offset: 0x0C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												OM117		OM116	
												rw		rw	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:2	OM117[1:0]	Pin OUTVALID output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description
1:0	OM116[1:0]	Pin IO3 output mode bit These bits are set and cleared by software. Refer to OM000[1:0] description

5.7.5. Port pull-up/down register0 (GPIO0_PUD0)

Address offset: 0x10

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD015[1:0]		PUD014[1:0]		PUD013[1:0]		PUD012[1:0]		PUD011[1:0]		PUD010[1:0]		PUD009[1:0]		PUD008[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD007[1:0]		PUD006[1:0]		PUD005[1:0]		PUD004[1:0]		Reserved		PUD002[1:0]		PUD001[1:0]		PUD000[1:0]	
rw		rw		rw		rw				rw		rw		rw	

Bits	Fields	Descriptions
31:30	PUD015[1:0]	Pin IO11 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description

29:28	PUD014[1:0]	Pin OE_EXT pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
27:26	PUD013[1:0]	Pin IO4 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
25:24	PUD012[1:0]	Pin IO5 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
23:22	PUD011[1:0]	Pin IO6 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
21:20	PUD010[1:0]	Pin LATCH_IN pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
19:18	PUD009[1:0]	Pin SYNC1_LATCH1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
17:16	PUD008[1:0]	Pin WD_STATE pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
15:14	PUD007[1:0]	Pin IO7 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
13:12	PUD006[1:0]	Pin IO8 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
11:10	PUD005[1:0]	Pin EOF pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
9:8	PUD004[1:0]	Pin SOF pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
7:6	Reserved	Must be kept at reset value
5:4	PUD002[1:0]	Pin IO18 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
3:2	PUD001[1:0]	Pin IO17 pull-up or pull-down bits

		These bits are set and cleared by software. Refer to PUD000[1:0] description
1:0	PUD000[1:0]	Pin IO16 pull-up or pull-down bits These bits are set and cleared by software. 00: Floating mode, no pull-up and pull-down (reset value) 01: With pull-up mode 10: With pull-down mode 11: analog mode

5.7.6. Port pull-up/down register1 (GPIO0_PUD1)

Address offset: 0x14

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PUD017[1:0]		PUD016[1:0]	
												rw		rw	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:2	PUD017[1:0]	Pin IO13 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
1:0	PUD016[1:0]	Pin IO12 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description

5.7.7. Port pull-up/down register2 (GPIO1_PUD0)

Address offset: 0x18

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUD115[1:0]		PUD114[1:0]		PUD113[1:0]		PUD112[1:0]		PUD111[1:0]		PUD110[1:0]		PUD109[1:0]		PUD108[1:0]	
rw		rw		rw		rw		rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUD107[1:0]		PUD106[1:0]		PUD105[1:0]		PUD104[1:0]		PUD103[1:0]		PUD102[1:0]		PUD101[1:0]		PUD100[1:0]	

rw	rw	rw	rw	rw	rw	rw	rw

Bits	Fields	Descriptions
31:30	PUD115[1:0]	Pin LINKACTLED0 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
29:28	PUD114[1:0]	Pin LINKACTLED1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
27:26	PUD113[1:0]	Pin EESIZE pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
25:24	PUD112[1:0]	Pin IRQ pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
23:22	PUD111[1:0]	Pin EESCL pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
21:20	PUD110[1:0]	Pin EESDA pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
19:18	PUD109[1:0]	Pin TESTMODE pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
17:16	PUD108[1:0]	Pin IO2 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
15:14	PUD107[1:0]	Pin IO1 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
13:12	PUD106[1:0]	Pin IO0 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
11:10	PUD105[1:0]	Pin WD_TRIG pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
9:8	PUD104[1:0]	Pin SYNC0_LATCH0 pull-up or pull-down bits These bits are set and cleared by software.

		Refer to PUD000[1:0] description
7:6	PUD103[1:0]	Pin IO9 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
5:4	PUD102[1:0]	Pin IO15 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD100[1:0] description
3:2	PUD101[1:0]	Pin IO14 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
1:0	PUD100[1:0]	Pin IO10 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description

5.7.8. Port pull-up/down register3 (GPIO1_PUD1)

Address offset: 0x1C

Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PUD117[1:0]		PUD116[1:0]	
												rw		rw	

Bits	Fields	Descriptions
31:4	Reserved	Must be kept at reset value.
3:2	PUD117[1:0]	Pin OUTVALID pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description
1:0	PUD116[1:0]	Pin IO3 pull-up or pull-down bits These bits are set and cleared by software. Refer to PUD000[1:0] description

5.7.9. EXMC control register (EXMC_CTL)

Address offset: 0x20

Reset value: 0x0000 0004

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													EXMCHSIZE[1:0]		EXMCTYPE
													rw		rw

Bits	Fields	Descriptions
31:3	Reserved	Must be kept at reset value
2:1	EXMCHSIZE[1:0]	EXMC hsize 00: 8-bit 01:16-bit 10:32-bit 11: reserved
0	EXMCTYPE	EXMC TYPE 0: 8-bit EXMC 1: 16-bit EXMC

6. TIMER

6.1. Basic Timer

6.1.1. Overview

The basic timer module has a 16-bit counter that can be used as an unsigned counter. The basic timer can be configured to generate interrupts. The resolution of basic timer is 100 μs.

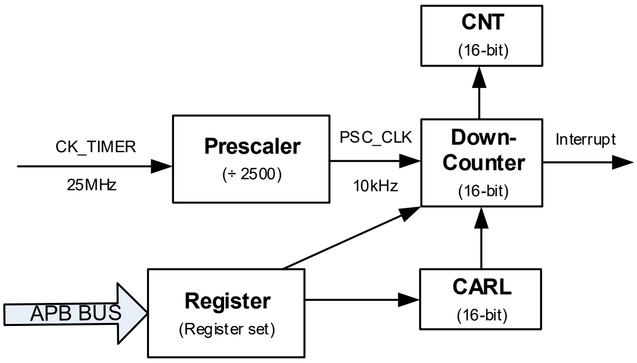
6.1.2. Characteristics

- Counter width: 16 bits.
- Source of count clock is internal clock only.
- Counter mode: count down.
- Resolution: 100 μs.
- Auto-reload function.
- Interrupt output: update event.

6.1.3. Block diagram

Figure 6-1. Basic timer block diagram provides details on the internal configuration of the basic timer.

Figure 6-1. Basic timer block diagram



6.1.4. Function overview

Clock source

The basic timer can only be clocked by the 25MHz internal timer clock, which is from the source named CK_TIMER in RCU.

The CK_TIMER will be divided by 2500 to generate the 10kHz counter clock (PSC_CLK). The resolution of the counter is 100 μs.

Down counting mode

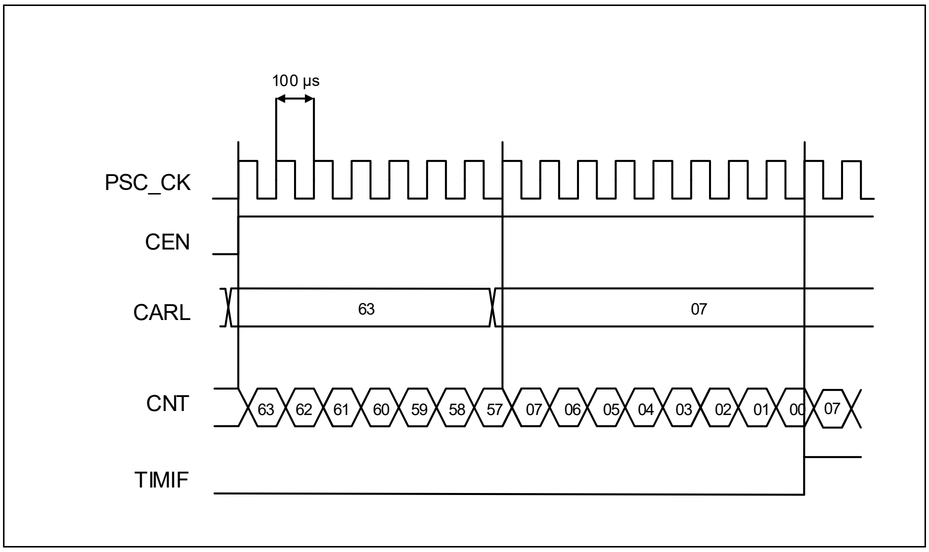
When the CEN bit in TIMERx_CTL0 register is set, The basic timer loads the TIMER_CNT register with a reload value in the CARL bit-filed of TIMER_CTL0 register. A new reload value can be written to the CARL bit-filed by software at any time. If the CEN bit is set, the TIMER_CNT register will be immediately loaded to the new reload value and continue to count down from that value.

If a chip-level reset occurs or the CEN bit in TIMER_CTL0 register is converted from 1 to 0, The CARL bit-filed is initialized to 0xFFFF. The CNT bit-filed is initialized to 0xFFFF on reset.

When the CEN bit in TIMER_CTL0 register is set, the counter counts down continuously from the counter reload value to 0. Once the counter reaches 0, the counter wraps around to 0xFFFF and the TIMIF bit in INTC_FLAG register is set. If TIMIE bit in INTC_EN register is set, basic timer generates the interrupt. The counter continues to count down from the reload value of CARL bit-filed. If the TIMIF bit in INTC_FLAG register is set, it can only be cleared by writing a 1 to the bit.

The following figure shows an example of the counter behavior when CARL bit-filed is converted from 0x63 to 0x07.

Figure 6-2. Timing chart of down counting mode



6.1.5. Registers definition

Basic Timer base address: 0x0000 3800

Control register 0 (TIMER_CTL0)

Address offset: 0x00
Reset value: 0x0000 FFFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		CEN	Reserved												
		rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CARL[15:0]															
rw															

Bits	Fields	Descriptions
31:30	Reserved	Must be kept at reset value.
29	CEN	Counter enable 0: Counter disable 1: Counter enable
28:16	Reserved	Must be kept at reset value.
15:0	CARL[15:0]	Counter auto reload value This bit-filed specifies the auto reload value of the counter. When the CEN bit is set, the counter counts down from this value. **Note:** When the CEN bit is converted from 1 to 0, this bit-filed wil be converted to 0xFFFF.

Counter register (TIMER_CNT)

Address offset: 0x04

Reset value: 0x0000 FFFF

This register has to be accessed by word (32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
r															

Bits	Fields	Descriptions
31:16	Reserved	Must be kept at reset value.
15:0	CNT[15:0]	This bit-filed indicates the current counter value.

6.2. Free-Running Counter (FRC)

6.2.1. Overview

The Free-Running Counter has a 32-bit counter that can be used as an unsigned counter.The counter clock is 25MHz.

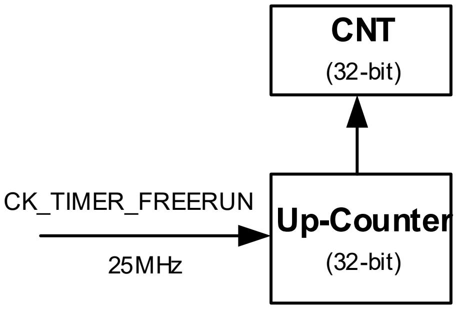
6.2.2. Characteristics

- Counter width: 32 bits.
- Source of count clock is internal clock only.
- Counter mode: count up.

6.2.3. Block diagram

Figure 6-3. FRC block diagram provides details on the internal configuration of the FRC.

Figure 6-3. FRC block diagram



6.2.4. Function overview

Clock source

The FRC can only be clocked by the 25MHz internal timer clock, which is from the source named CK_TIMER_FREERUN in RCU.

Up counting mode

The counter counts up continuously from 0 to 0xFFFFFFFF and the counter frequency is 25MHz. Once the counter reaches the maximum, the counter recounts from 0. The FRC does not generate interrupts. If a chip-level reset occurs, the counter is initialized to 0.

The current count value can be read from FRC_CNT register.

The counter can take up to 160 ns to clear after a reset event.

6.2.5. Registers definition

Free-Running Counter base address: 0x0000 3808

Counter Register (FRC_CNT)

Address offset: 0x00
Reset value: 0x0000 0000

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT [31:16]															
r															

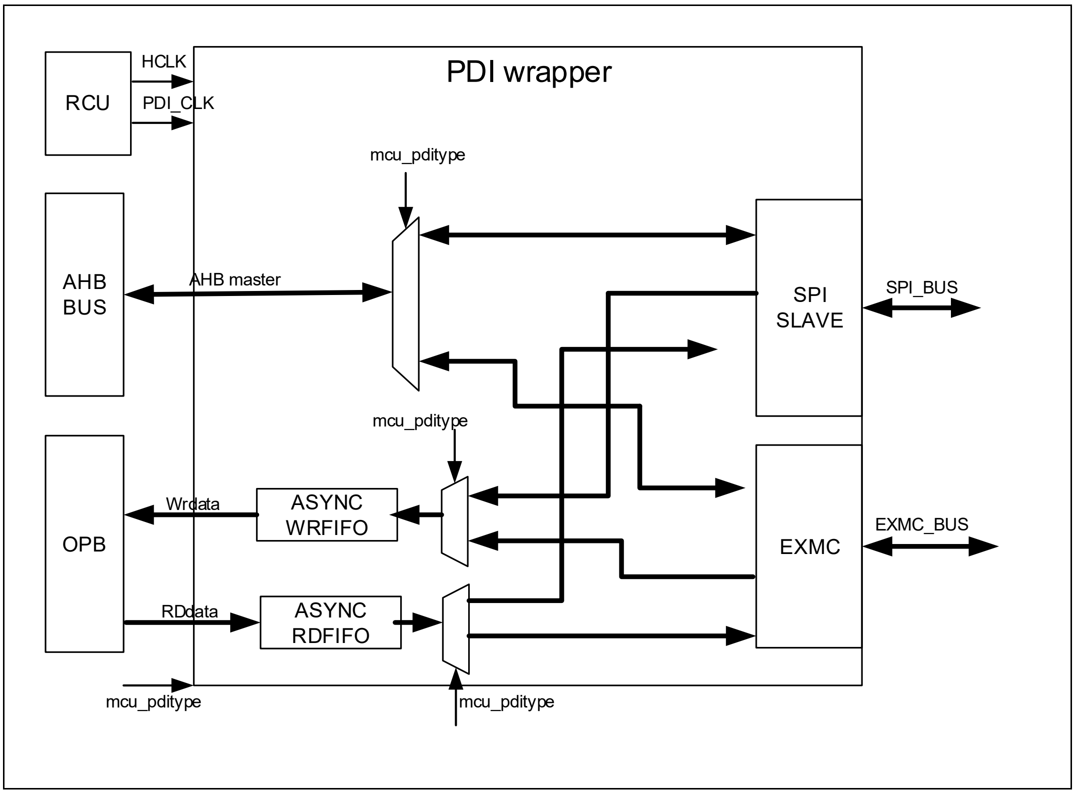
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT [15:0]															
r															

Bits	Fields	Descriptions
31:0	CNT[31:0]	This bit-filed indicates the current counter value. If a chip-level reset occurs, the bit-filed is initialized to 0 and the counter counts up from 0. Once the counter reaches the maximum, the bit-filed is converted to 0 and the counter continues to count up from 0. **Note:** The counter can take up to 160 ns to clear after a reset event.

7. PDI Wrapper

In the GD EtherCAT, EXMC and SPI SLAVE are packaged into a wrapper for system integration. The PDI wrapper Used for data selection between SPI and EXMC. Internal integration of two asynchronous FIFOs, both 16X32bit. The SPI SLAVE and EXMC have only one work at the same time, which is selected by the pad of MCU_PDITYPE.

Figure 7-1. Block diagram of PDI wrapper



Only one of SPI SLAVE and EXMC is valid at the same time. SPI SLAVE or EXMC accesses registers through AHB channel and accesses CORE ram data through ASYNC RDFIFO and ASYNC WRFIFO. EXMC also has the same data path as SPI SLAVE.

The pad of MCU_PDITYPE select the working access interface module. When the pad of MCU_PDITYPE is 0, only the SPI SLAVE can access internal data. When the pad of MCU_PDITYPE is 1, only EXMC can access internal data.

PDI_CLK provides the clock for SPI_SLAVE, EXMC, and ASYNC_FIFO. When the pad of MCU_PDITYPE is 0, PDI_CLK comes from SPI_SCK. When the pad of MCU_PDITYPE is 1, PDI_CLK comes from EXMC_CLK. HCLK is a 100MHz system clock that provides clocks for ASYNC_FIFO, SPI_SLAVE, and EXMC.

7.1. SPI / QSPI / OSPI slave

7.1.1. Overview

The EtherCAT support SPI / QSPI / OSPI slave module.

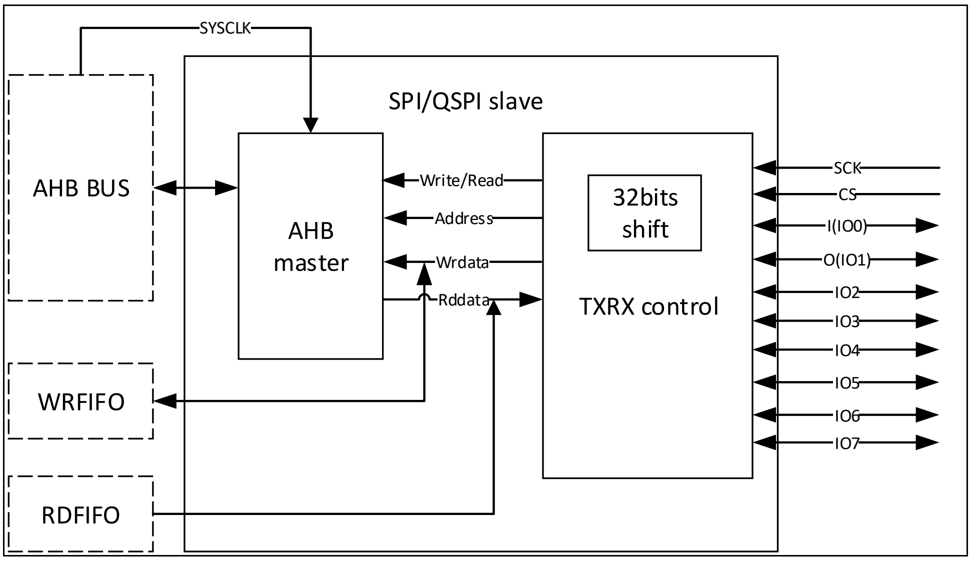
7.1.2. Characteristics

- Supports a maximum SPI clock rate of 100Mhz.
- Only support slave mode.
- All samples are sampled along the rising edge.
- Supports FIFO buffer access.

7.1.3. Block diagram

The block diagram of SPI is shown in *Figure 7-2. Block diagram of SPI*.

Figure 7-2. Block diagram of SPI



7.1.4. SPI signal description

Pin description

The SPI / QSPI slave module contain 2 kinds of pin mode: 4-wire mode and 6-wire mode. All modes contain common pins, SCK and CS.

Table 7-1. 4-wire mode

Pin name	Description
SCK	SPI CLK
CS	Slave select signal
I	Input pin, receive SPI / QSPI master data
O	Output pin, transmit data to SPI / QSPI master data

Table 7-2. 6-wire mode

Pin name	Description
SCK	SPI / QSPI clock

CS	Slave select signal
SIO0	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO1	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO2	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO3	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data

Table 7-3. OSPI 8-line mode

Pin name	Description
SCK	SPI / QSPI clock
CS	Slave select signal
SIO0	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO1	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO2	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO3	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO4	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO5	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO6	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data
SIO7	Inout pin, receive SPI / QSPI master data and transmit data to SPI / QSPI master data

7.1.5. SPI/QSPI/OSPI slave controller

Function overview

The SPI slave interface can access registers and FIFOs with fewer pins. Single, Dual and Quad bit lanes are supported in SPI mode with a clock rate of up to 100 MHz. QSPI mode always uses four bit lanes and also operates at up to 80 MHz.OSPI mode always uses eight bit lanes and also operates at up to 80MHz.

Function description

The following is an overview of the functions provided by the SPI/ QSPI/OSPI Client:

- Fast Read: 4-wire (clock, select, data in and data out) reads at up to 80 MHz. Serial command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- Dual / Quad Output Read: 4 or 6-wire (clock, select, data in / out) reads at up to 80 MHz. Serial command and address, parallel data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- Dual / Quad I/O Read: 4 or 6-wire (clock, select, data in / out) reads at up to 80 MHz. Serial command, parallel address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- QSPI Read: 6-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- OSPI Read: 10-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- Write: 4-wire (clock, select, data in and data out) writes at up to 80 MHz. Serial command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- Dual / Quad Data Write: 4 or 6-wire (clock, select, data in / out) writes at up to 80 MHz. Serial command and address, parallel data. Single and multiple register writes with incrementing, decrementing or static addressing.
- Dual / Quad Address / Data Write: 4 or 6-wire (clock, select, data in / out) writes at up to 80 MHz. Serial command, parallel address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- QSPI Write: 6-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- OSPI Write: 10-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.

Operation description

Input data on the IO [7:0] pins is sampled on the rising edge of the SCK input clock. Output data is sourced on the IO [7:0] pins with the falling edge of the clock. The SCK input clock can be either an active high pulse or an active low pulse. When the CS chip select input is high, the IO [7:0] inputs are ignored and the IO [7:0] outputs are three stated.

In SPI mode, the 8-bit instruction is started on the first rising edge of the input clock after CS goes active. The instruction is always input serially on I / IO0.

For read and write instructions, two address bytes follow the instruction byte. Depending on the instruction, the address bytes are input either serially, or 2 or 4 bits per clock. Although all registers are accessed as DWORDs, the address field is considered a byte address. bits 15 and 14 of the address field specifies that address is auto-decremented (10b) or auto-incremented (01b) for continuous accesses. (if accessing inner fifo, bits 15 and 14 will be ignored)

For all read instructions, dummy byte cycles follow the address bytes. The device does not drive the outputs during the dummy byte cycles. The dummy byte(s) are input either serially, or 2 or 4 or 8 bits per clock. The data is input either serially, or 2 or 4 bits per clock.

For read and write instructions, one or more 32-bit data fields follow the dummy bytes (if present, else they follow the address bytes). The data is input either serially, or 2 or 4 or 8 bits per clock.

QSPI mode is entered from SPI with the Enable Quad I/O (EQIO) instruction. Once in QSPI mode, all further command, addresses, dummy bytes and data bytes are 4 bits per clock. QSPI mode can be exited using the Reset Quad I/O (RSTQIO) instruction.

OSPI mode is entered from SPI with the Enable Octa I/O (EOIO) instruction. Once in OSPI mode, all further command, addresses, dummy bytes and data bytes are 8 bits per clock. OSPI mode can be exited using the Reset Octa I/O (RSTQIO) instruction

All instructions, addresses and data are transferred with the most-significant bit (msb) or di-bit (msd) or nibble (msn) first. Addresses are transferred with the most-significant byte (MSB) first. Data is transferred with the least-significant byte (LSB) first (little endian).

The SPI interface supports up to a 100 MHz input clock. (exception: for the QSPI instruction, the number of accessed data bytes is 4 with 100Mhz. if master want to access more data bytes, master could use lower speed (less than or equal to 60Mhz))

The SPI interface supports a minimum time of 50 ns between successive commands (a minimum CS inactive time of 50 ns).

The instructions supported in SPI mode are listed in ***Table 7-4. SPI instructions***. QSPI instructions are listed in ***Table 7-5. QSPI instruction*****Table 7-5. QSPI instruction**. Unsupported instructions are must not be used.

Table 7-4. SPI instructions

Instruction	Description	Bit width	Inst code	Addr Bytes	Dummy Bytes	Data bytes	Max Freq
Configuration							
EQIO	Enable QSPI	1-0-0	38h	0	0	0	100Mhz
EOIO	Enable OSPI	1-0-0	3Ah	0	0	0	100Mhz
RSTIO	Reset SPI	1-0-0	FFh	0	0	0	100Mh

							z
Read							
READ	Read	1-1-1	0Bh	2	1	4 to ∞	100Mhz
SDOR	SPI Dual Output Read	1-1-2	3Bh	2	1	4 to ∞	100Mhz
SDIOR	SPI Dual I/O Read	1-2-2	BBh	2	2	4 to ∞	100Mhz
SQOR	SPI Quad Output Read	1-1-4	6Bh	2	1	4 to ∞	100Mhz
SQIOR	SPI Quad I/O Read	1-4-4	EBh	2	4	4 to ∞	100Mhz
Write							
WRITE	Write	1-1-1	02h	2	0	4 to ∞	100Mhz
SDDW	SPI Dual Data Write	1-1-2	32h	2	0	4 to ∞	100Mhz
SDADW	SPI Dual Address / Data Write	1-2-2	B2h	2	0	4 to ∞	100Mhz
SQDW	SPI Quad Data Write	1-1-4	62h	2	0	4 to ∞	100Mhz
SQADW	SPI Quad Address / Data Write	1-4-4	E2h	2	0	4 to ∞	100Mhz

Table 7-5. QSPI instruction

Instruction	Description	Bit width	Inst code	Addr Bytes	Dummy Bytes	Data bytes	Max Freq
Configuration							
RSTQIO	Reset QSPI	4-0-0	FFh	0	0	4 to ∞	100Mhz
Read							
READ	Read	4-4-4	0Bh	2	3	4 to ∞	100Mhz
Write							
WRITE	write	4-4-4	02h	2	0	4 to ∞	100Mhz

Table 7-6. OSPI instruction

Instruction	Description	Bit width	Inst code	Addr Bytes	Dummy Bytes	Data bytes	Max Freq
Configuration							
RSTOIO	Reset OSPI	8-0-0	FFh	0	0	4 to ∞	100Mhz
Read							
READ	Read	8-8-8	0Bh	2	8	4 to ∞	100Mhz
Write							
WRITE	write	8-8-8	02h	2	0	4 to ∞	100Mhz

Note: The bit width format is: command bit width, address / dummy bit width, data bit width.

For example, 1-2-4 means command uses 1 line, address/dummy uses 2 lines, data uses 4 lines.

SPI configuration commands

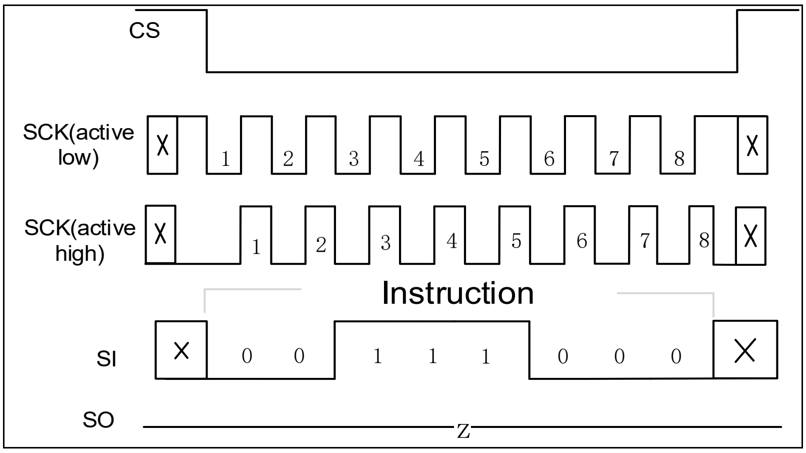
Enable QSPI

The enable QSPI instruction changes the mode of operation to QSPI. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

The SPI client interface is selected by first bringing CS active. The 8-bit EQIO instruction, 38h, is input into the I / IO [0] pin one bit per clock. The CS input is brought inactive to conclude the cycle.

Figure 7-3. Enable QSPI illustrates the Enable QSPI instruction.

Figure 7-3. Enable QSPI



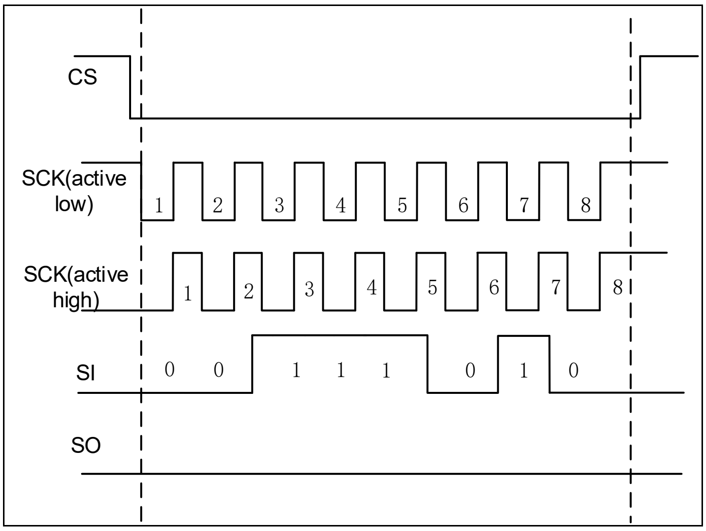
Enable OSPI

The Enable OSPI instruction changes the mode of operation to OSPI. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in OSPI bus protocol.

The SPI client interface is selected by first bringing CS active. The 8-bit EOIO instruction, 3Ah, is input into the I / IO[0] pin one bit per clock. The CS input is brought inactive to conclude the cycle.

Figure 7-4. Enable OSPI illustrates the Enable OSPI instruction.

Figure 7-4. Enable OSPI



Reset QSPI

The Reset QSPI / OSPI instruction changes the mode of operation to SPI. This instruction is supported in SPI / QSPI / OSPI bus protocols with clock frequencies up to 80 MHz.

The SPI / QSPI / OSPI client interface is selected by first bringing CS active. The 8-bit RSTQIO instruction, FFh, is input into the I / IO[0] pin, one bit per clock, in SPI mode and into the IO[3:0] pins, four bits per clock, in QSPI mode. The CS input is brought inactive to conclude the cycle.

Figure 7-5. SPI MODE RESET SPI illustrates the Reset SPI instruction for SPI mode.

Figure 7-6. QSPI MODE RESET QSPI illustrates the Reset QSPI instruction for QSPI mode.

Figure 7-7. OSPI MODE RESET OSPI illustrates the Reset OSPI instruction for OSPI mode.

Figure 7-5. SPI MODE RESET SPI

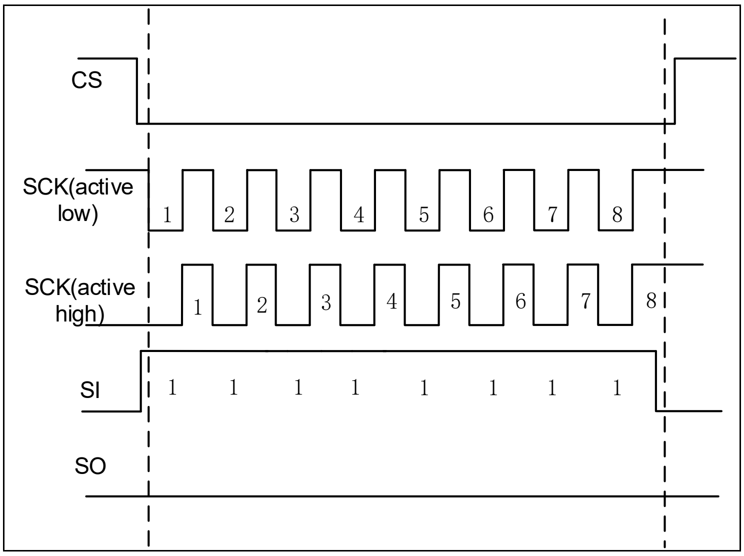


Figure 7-6. QSPI MODE RESET QSPI

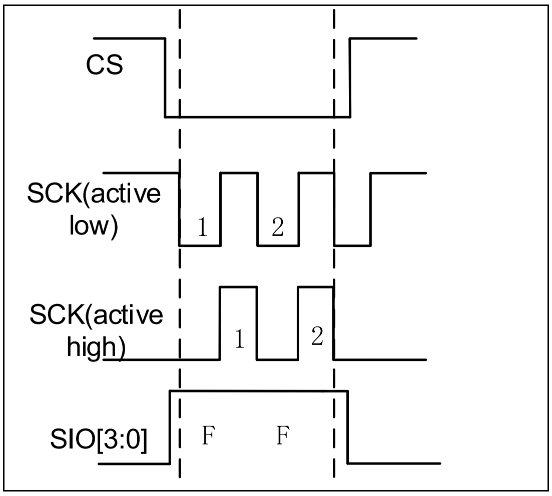
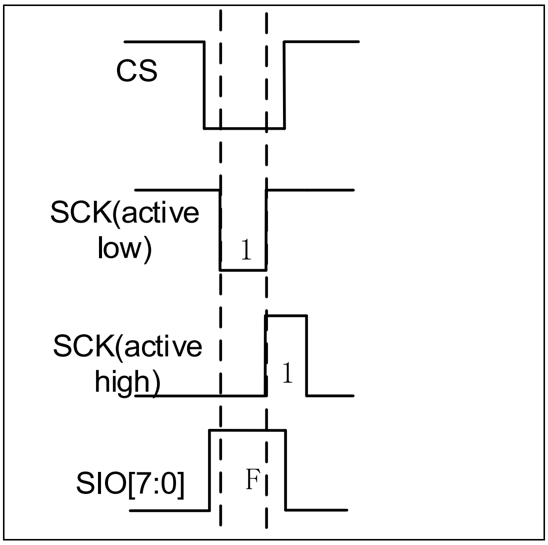


Figure 7-7. OSPI MODE RESET OSPI



SPI READ COMMANDS

Various read commands are support by the SPI / QSPI client. The following applies to all read commands.

MULTIPLE READS

Additional reads, beyond the first, are performed by continuing the clock pulses while CS is active. The upper two bits of the address specify auto-incrementing (address[15:14]=01b) or auto-decrementing (address[15:14]=10b). The internal DWORD address is incremented, decremented, or maintained based on these bits. Maintaining a fixed internal address is useful for register polling.

READ

The Read instruction inputs the instruction code and the address and dummy bytes one bit per clock and outputs the data one bit per clock. In QSPI mode, the instruction code and the address and dummy bytes are input four bits per clock and the data is output four bits per clock. This instruction is supported in SPI and QSPI bus protocols with clock frequencies up to 80 MHz.

The SPI/QSPI/OSPI client interface is selected by first bringing CS active. For SPI mode, the 8-bitREAD instruction, 0Bh, is input into the I / IO [0] pin, followed by the two address bytes and 1 dummy byte. For QSPI mode, the 8-bit FASTREAD instruction is input into the IO [3:0] pins, followed by the two address bytes and 3 dummy bytes. The address bytes specify a BYTE address within the device. For OSPI mode, the 8-bit FASTREAD instruction is input into the IO [7:0] pins, followed by the two address bytes and 8 dummy bytes. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy bit (or nibble), the O / IO [1] pin is driven starting with the msb of the LSB of the selected register. For QSPI mode, IO [3:0] are driven starting with the msn of the LSB of the selected register. For OSPI mode, IO [7:0] are driven starting with the msn of the LSB of the selected register. The remaining register bits are shifted out on subsequent falling clock edges.

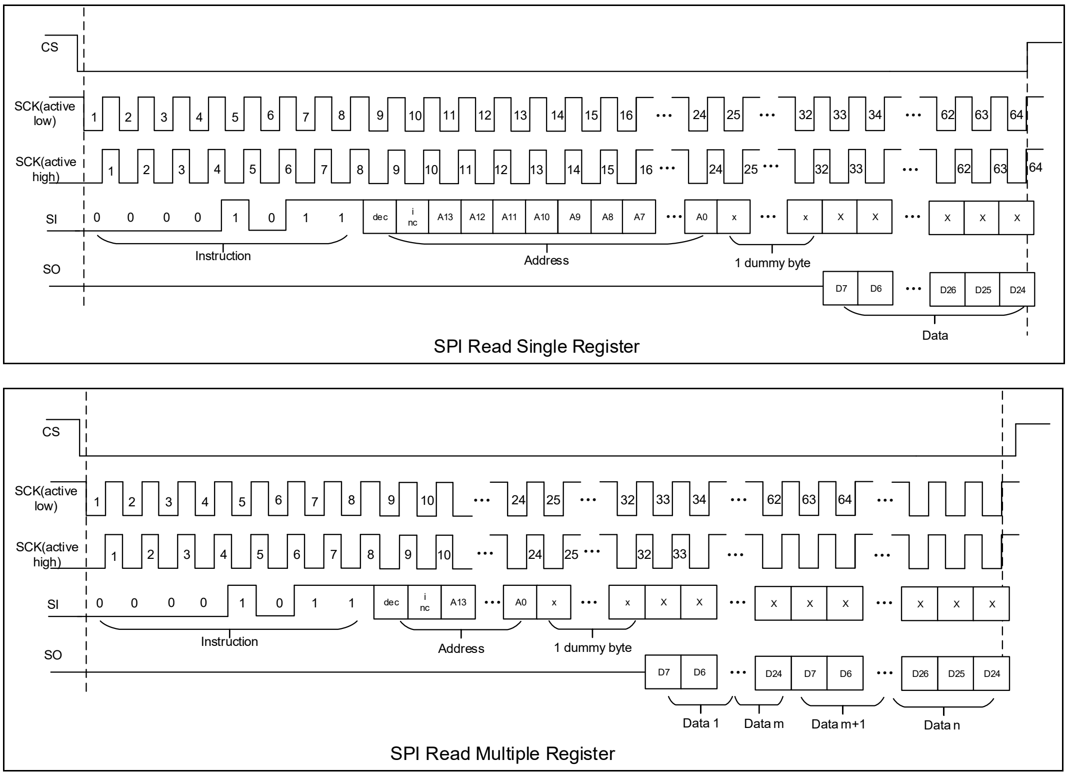
The CS input is brought inactive to conclude the cycle. The O / IO [7:0] pins are three-stated at this time.

Figure 7-8. SPI READ illustrates a typical single and multiple register fast read for SPI mode.

Figure 7-9. QSPI READ illustrates a typical single and multiple register fast read for QSPI mode.

Figure 7-10. OSPI READ illustrates a typical single and multiple register fast read for OSPI mode.

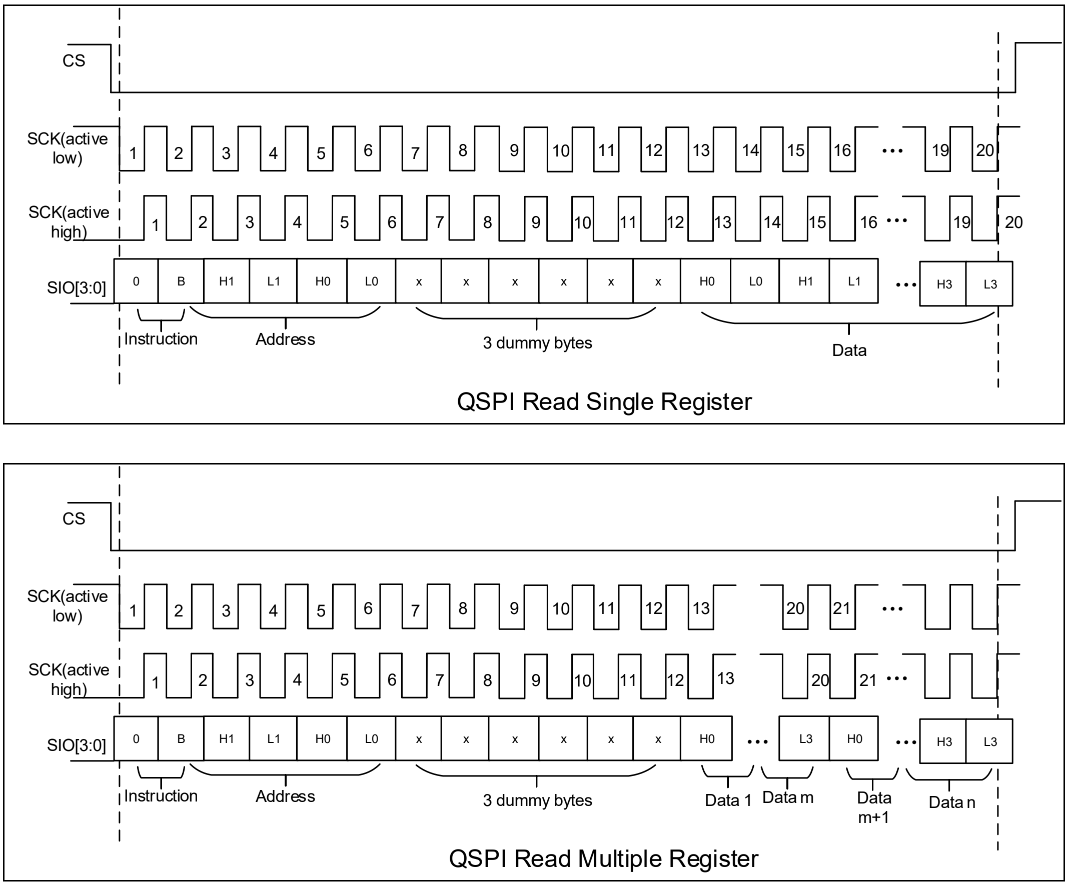
Figure 7-8. SPI READ



SPI Read Single Register

SPI Read Multiple Register

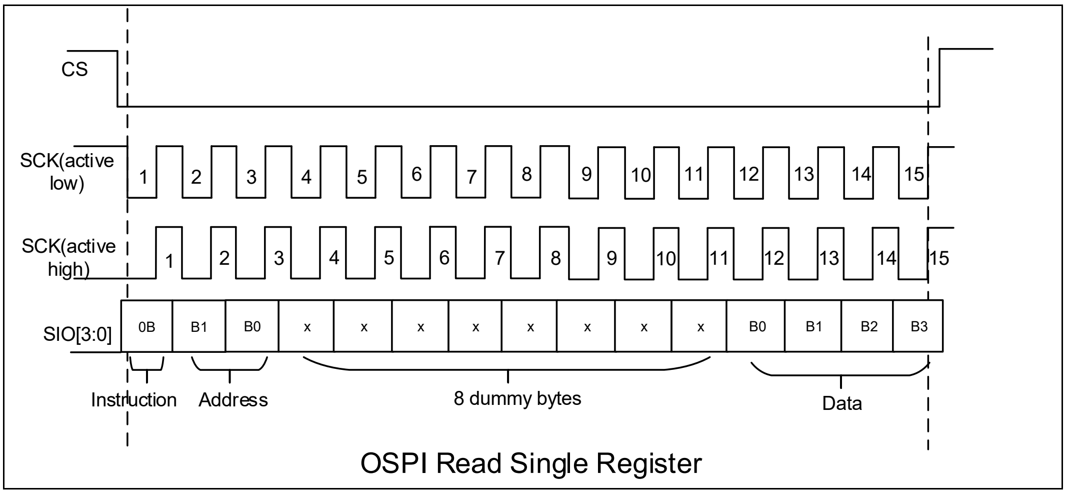
Figure 7-9. QSPI READ



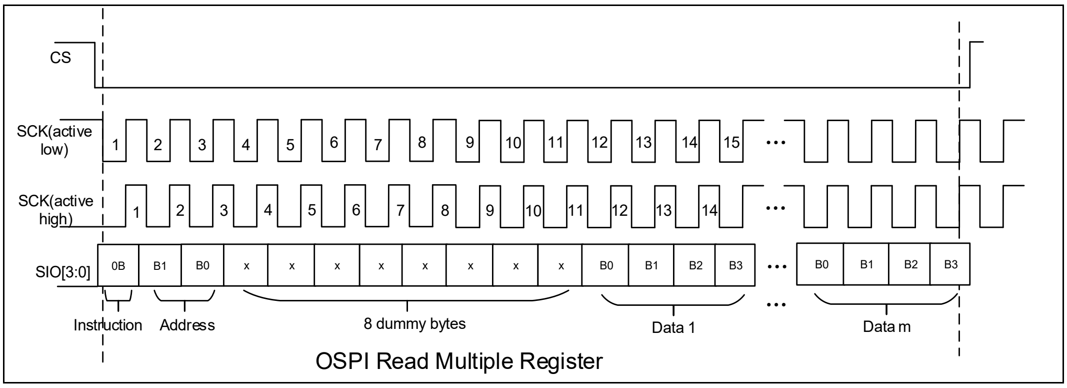
QSPI Read Single Register

QSPI Read Multiple Register

Figure 7-10. OSPI READ



OSPI Read Single Register



OSPI Read Multiple Register

Dual Output Read

The SPI Dual Output Read instruction inputs the instruction code and the address and dummy bytes one bit per clock and outputs the data two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

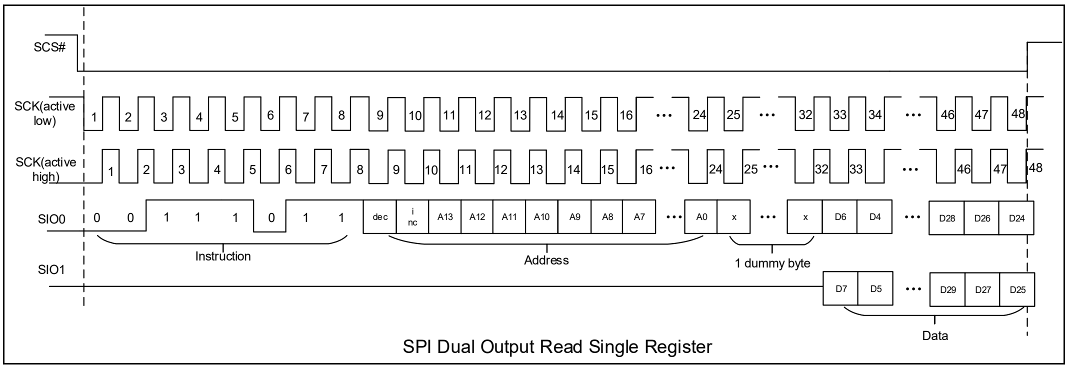
The SPI client interface is selected by first bringing CS active. The 8-bit SDOR instruction, 3Bh, is input into the IO [0] pin, followed by the two address bytes and 1 dummy byte. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy di-bit, the IO [1:0] pins are driven starting with the msbs of the LSB of the selected register. The remaining register di-bits are shifted out on subsequent falling clock edges.

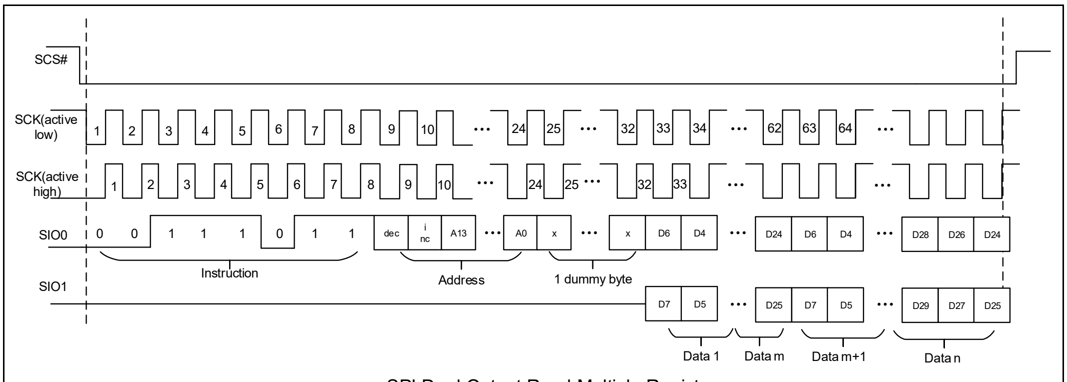
The CS input is brought inactive to conclude the cycle. The IO [1:0] pins are three-stated at this time.

Figure 7-11. SPI DUAL OUTPUT READ illustrates a typical single and multiple register dual output read.

Figure 7-11. SPI DUAL OUTPUT READ



SPI Dual Output Read Single Register



SPI Dual Output Read Multiple Register

QUAD Output Read

The SPI Quad Output Read instruction inputs the instruction code and the address and dummy bytes one bit per clock and outputs the data four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

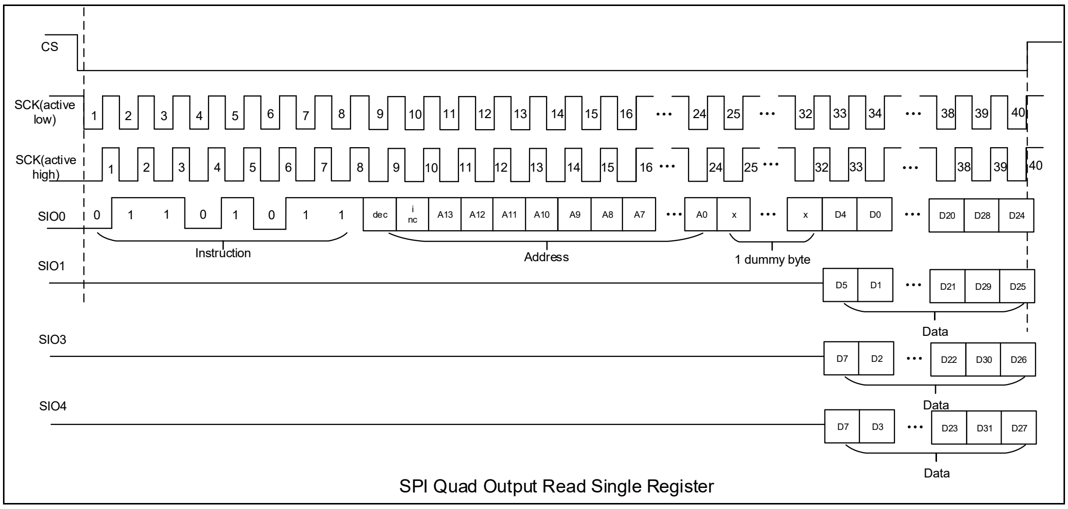
The SPI client interface is selected by first bringing CS active. The 8-bit SQOR instruction, 6Bh, is input into the IO [0] pin, followed by the two address bytes and 1 dummy byte. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy bit, the IO [3:0] pins are driven starting with the msn of the LSB of the selected register. The remaining register nibbles are shifted out.

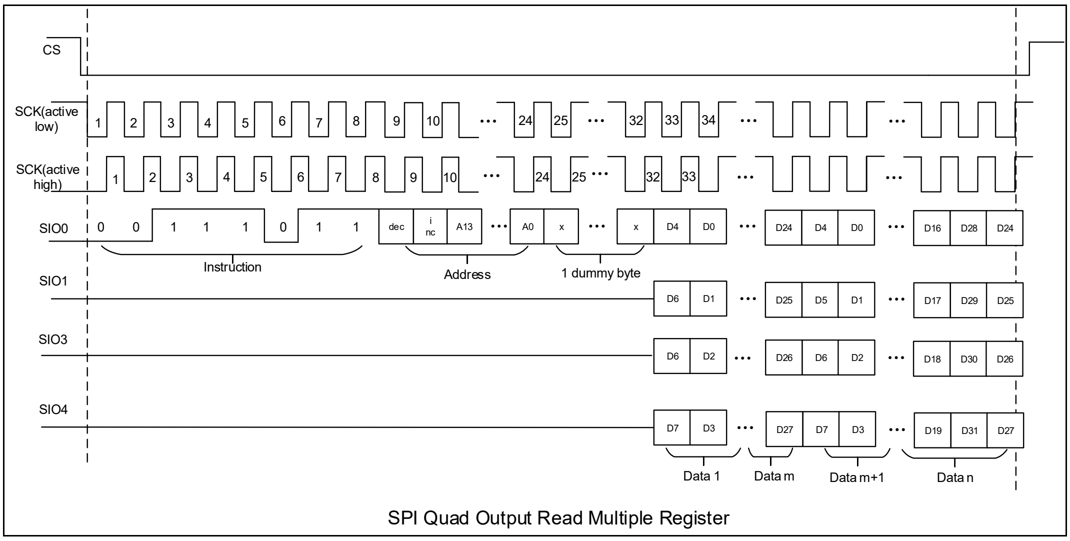
The CS input is brought inactive to conclude the cycle. The IO [3:0] pins are three-stated at this time.

Figure 7-12. SPI QUAD OUTPUT READ illustrates a typical single and multiple register quad output read.

Figure 7-12. SPI QUAD OUTPUT READ



SPI Quad Output Read Single Register



SPI Quad Output Read Multiple Register

Dual I/O Read

The SPI Dual I/O Read instruction inputs the instruction code one bit per clock and the address and dummy bytes two bits per clock and outputs the data two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

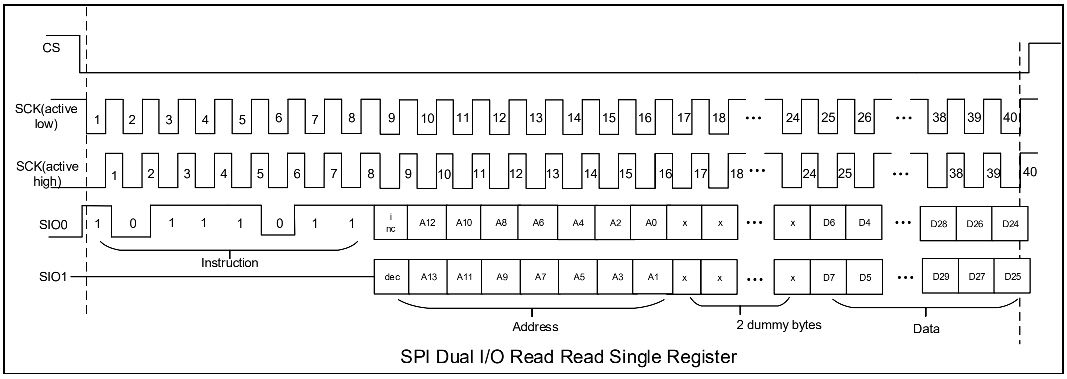
The SPI client interface is selected by first bringing CS active. The 8-bit SDIOR instruction, BBh, is input into the IO [0] pin, followed by the two address bytes and 2 dummy bytes into the IO [1:0] pins. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy di-bit, the IO [1:0] pins are driven starting with the msbs of the LSB of the selected register. The remaining register di-bits are shifted out on subsequent falling clock edges.

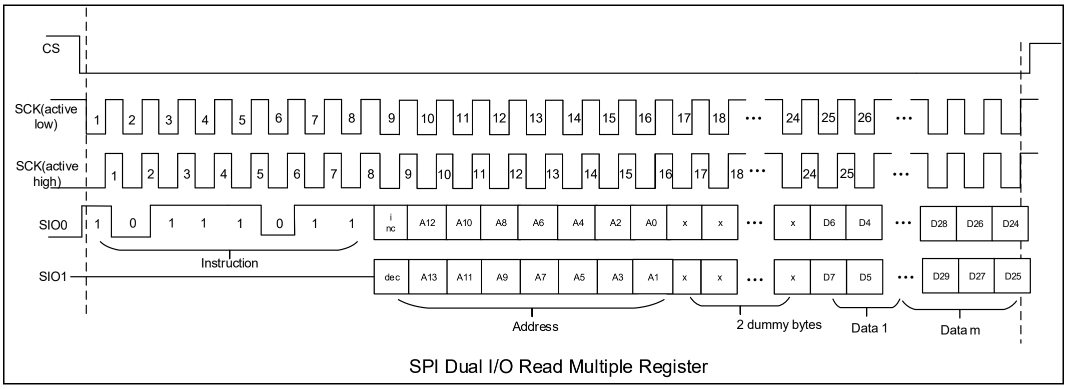
The CS input is brought inactive to conclude the cycle. The IO [1:0] pins are three-stated at this time.

Figure 7-13. SPI DUAL I/O READ illustrates a typical single and multiple register dual I/O read.

Figure 7-13. SPI DUAL I/O READ



SPI Dual I/O Read Read Single Register


CS
SCK(active low)
SCK(active high)
SIO0
SIO1
1 0 1 1 1 0 1 1
Instruction
i nc A12 A10 A8 A6 A4 A2 A0 x x ... x D6 D4 ... D28 D26 D24
dec A13 A11 A9 A7 A5 A3 A1 x x ... x D7 D5 ... D29 D27 D25
Address
2 dummy bytes
Data 1
Data m
SPI Dual I/O Read Multiple Register

Quad I/O Read

The SPI Quad I/O Read instruction inputs the instruction code one bit per clock and the address and dummy bytes four bits per clock and outputs the data four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

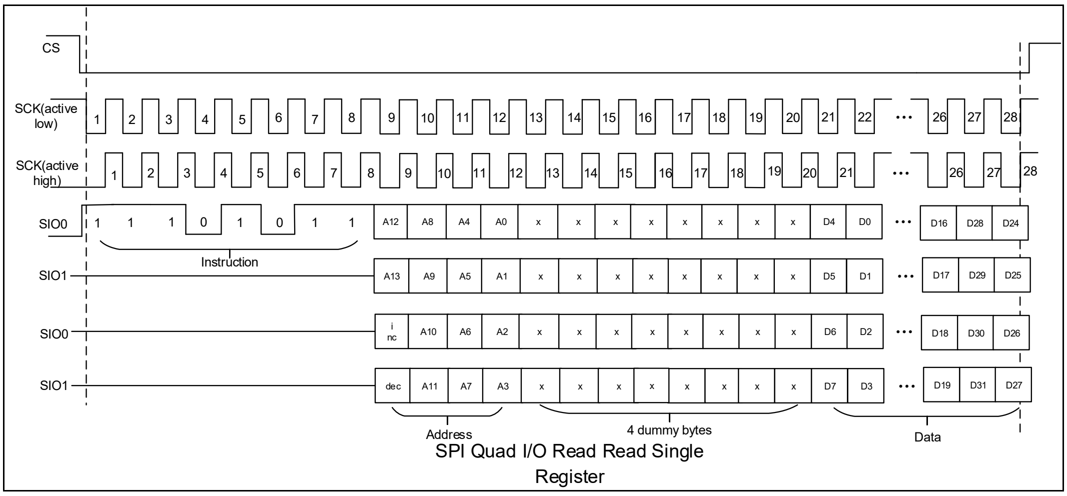
The SPI client interface is selected by first bringing CS active. The 8-bit SQIOR instruction, EBh, is input into the IO [0] pin, followed by the two address bytes and 4 dummy bytes into the IO [3:0] pins. The address bytes specify a BYTE address within the device.

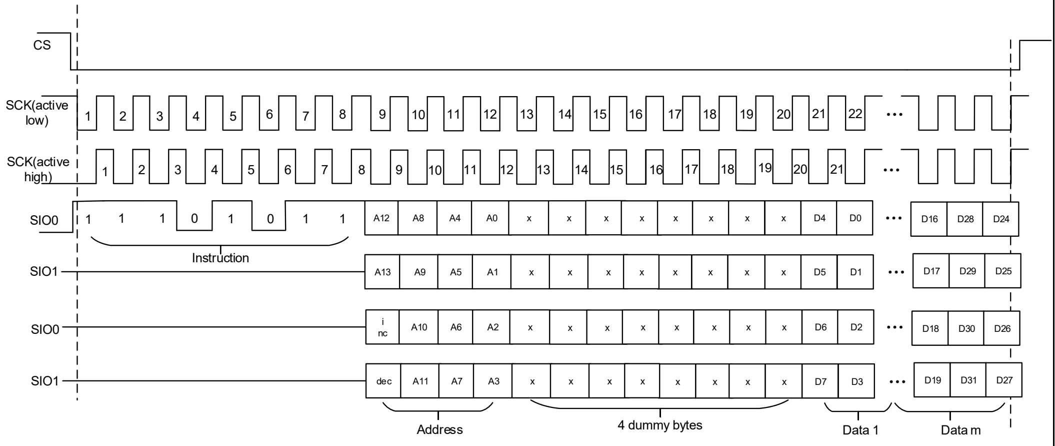
On the falling clock edge following the rising edge of the last dummy nibble, the IO [3:0] pins are driven starting with the msn of the LSB of the selected register. The remaining register nibbles are shifted out on subsequent falling clock edges.

The CS input is brought inactive to conclude the cycle. The IO [3:0] pins are three-stated at this time.

Figure 7-14. SPI QUAD I/O READ illustrates a typical single and multiple register dual output read.

Figure 7-14. SPI QUAD I/O READ


CS
SCK(active low)
SCK(active high)
SIO0
SIO1
SIO0
SIO1
1 1 1 0 1 0 1 1
Instruction
A12 A8 A4 A0 x x x x x x x x D4 D0 ... D16 D28 D24
A13 A9 A5 A1 x x x x x x x x D5 D1 ... D17 D29 D25
i nc A10 A6 A2 x x x x x x x x D6 D2 ... D18 D30 D26
dec A11 A7 A3 x x x x x x x x D7 D3 ... D19 D31 D27
Address
4 dummy bytes
Data
SPI Quad I/O Read Read Single Register



SPI Quad I/O Read Multiple Register

SPI WRITE COMMANDS

Multiple write commands are support by the SPI/QSPI client. The following applies to all write commands.

MULTIPLE WRITES

Multiple reads are performed by continuing the clock pulses and input data while CS is active. The upper two bits of the address specify auto-incrementing (address[15:14]=01b) or auto-decrementing (address[15:14]=10b). The internal DWORD address is incremented, decremented, or maintained based on these bits. Maintaining a fixed internal address may be useful for register “bit-banging” or other repeated writes.

Write

The Write instruction inputs the instruction code and address and data bytes one bit per clock. In QSPI mode, the instruction code and the address and data bytes are input four bits per clock. This instruction is supported in SPI and QSPI bus protocols with clock frequencies up to 80 MHz.

The SPI/QSPI client interface is selected by first bringing CS active. For SPI mode, the 8-bit WRITE instruction, 02h, is input into the I / IO [0] pin, followed by the two address bytes. For QSPI mode, the 8-bit WRITE instruction, 02h, is input into the IO [3:0] pins, followed by the two address bytes. For OSPI mode, the 8-bit WRITE instruction, 02h, is input into the IO [7:0] pins, followed by the two address bytes. The address bytes specify a BYTE address within the device.

The data follows the address bytes. For SPI mode, the data is input into the I / IO [0] pin starting with the msb of the LSB. For QSPI mode the data is input nibble wide using IO [3:0] starting with the msn of the LSB. For OSPI mode the data is input nibble wide using IO [7:0] starting with the msn of the LSB. The remaining bits/ nibbles are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the CS is returned high, the write is considered invalid and the register is not affected.

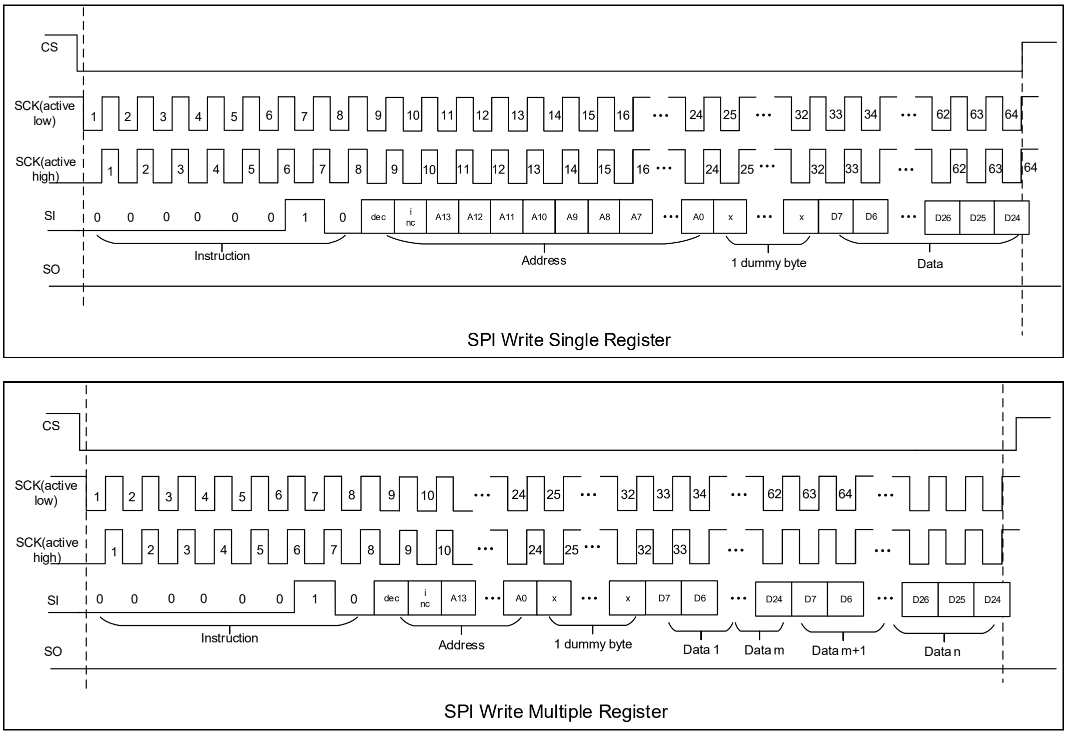
The CS input is brought inactive to conclude the cycle.

Figure 7-15. SPI WRITE illustrates a typical single and multiple register write for SPI mode.

Figure 7-16. QSPI WRITE illustrates a typical single and multiple register write for QSPI mode.

Figure 7-17. OSPI WRITE illustrates a typical single and multiple register write for OSPI mode.

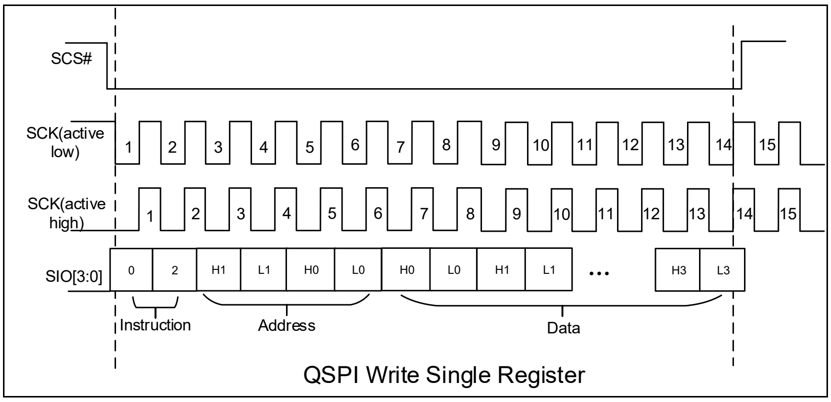
Figure 7-15. SPI WRITE



SPI Write Single Register

SPI Write Multiple Register

Figure 7-16. QSPI WRITE



QSPI Write Single Register

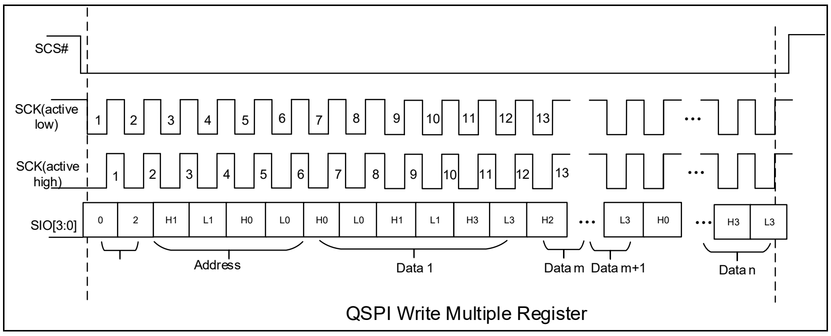
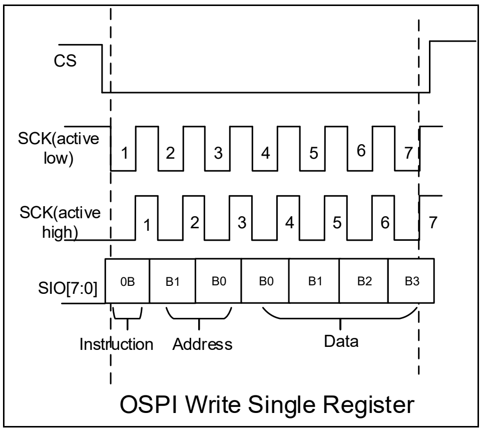
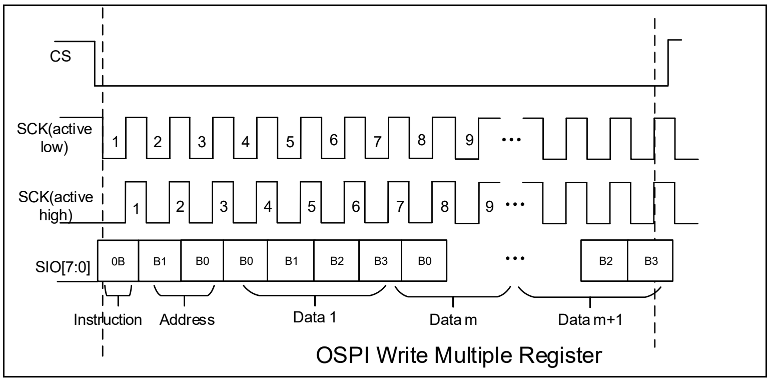


Figure 7-17. OSPI WRITE





Dual Data Read

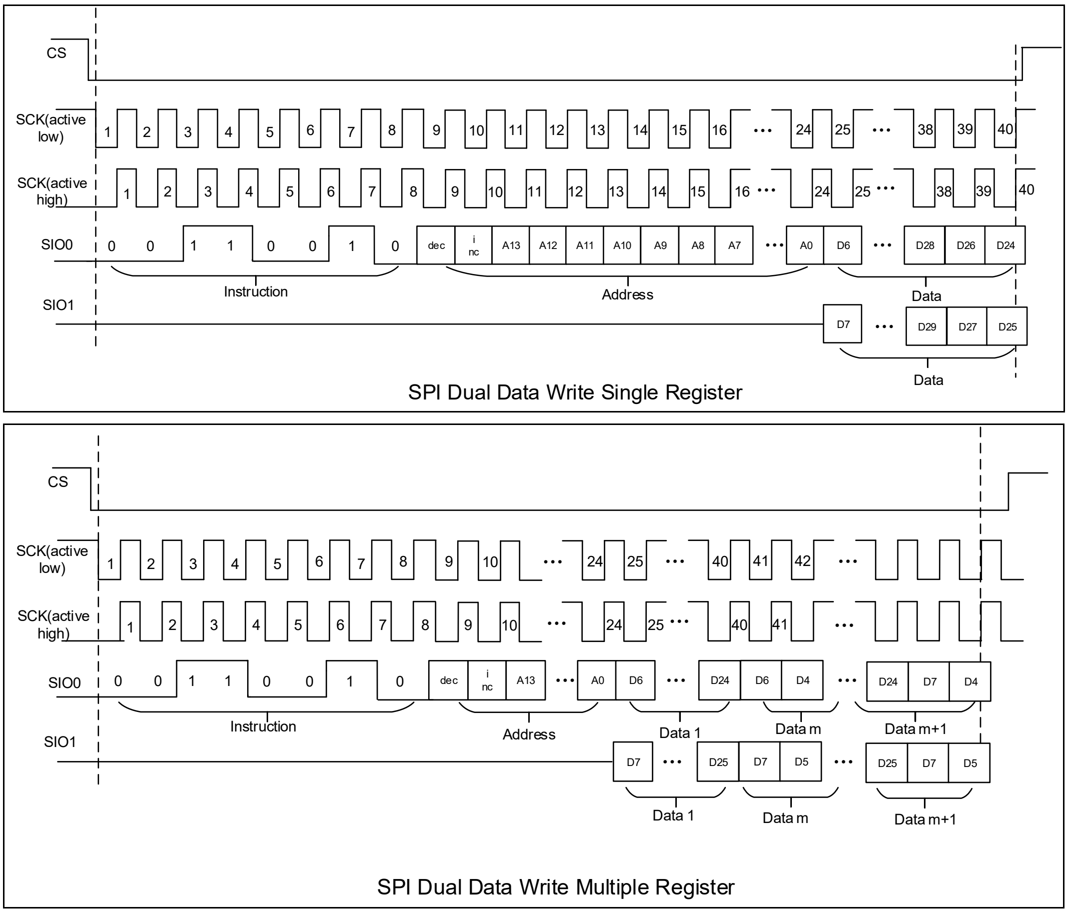
The SPI Dual Data Write instruction inputs the instruction code and address bytes one bit per clock and inputs the data two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

The SPI client interface is selected by first bringing CS active. The 8-bit SDDW instruction, 32h, is input into the IO [0] pin, followed by the two address bytes. The address bytes specify a BYTE address within the device. The data follows the address bytes. The data is input into the IO [1:0] pins starting with the msbs of the LSB. The remaining di-bits are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the CS is returned high, the write is considered invalid and the register is not affected.

The CS input is brought inactive to conclude the cycle.

Figure 7-18. SPI DUAL DATA WRITE illustrates a typical single and multiple register dual data write.

Figure 7-18. SPI DUAL DATA WRITE



SPI Dual Data Write Single Register

SPI Dual Data Write Multiple Register

Quad Data Read

The SPI Quad Data Write instruction inputs the instruction code and address bytes one bit per clock and inputs the data four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

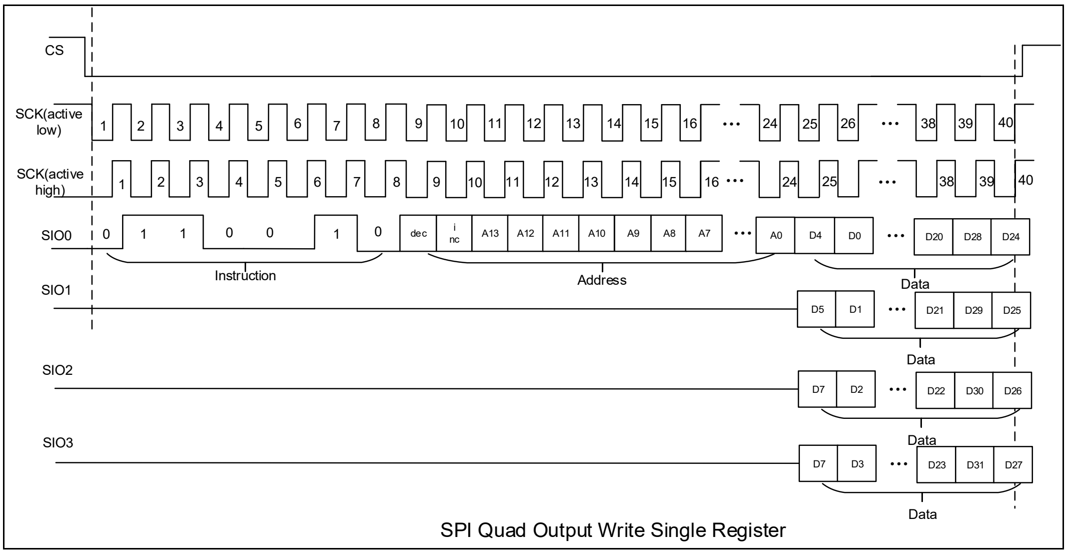
The SPI slave interface is selected by first bringing CS active. The 8-bit SQDW instruction, 62h, is input into the IO[0] pin, followed by the two address bytes. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the IO[3:0] pins starting with the msn of the LSB. The remaining nibbles are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the CS is returned high, the write is considered invalid and the register is not affected.

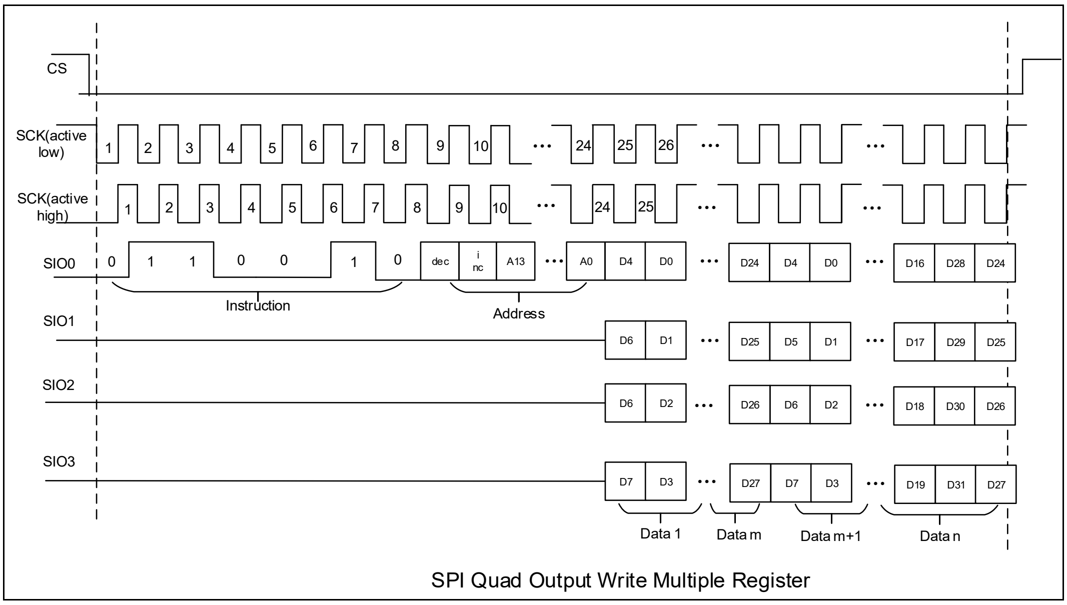
The CS input is brought inactive to conclude the cycle.

Figure 7-19. SPI QUAD DATA WRITE illustrates a typical single and multiple register quad data write.

Figure 7-19. SPI QUAD DATA WRITE



SPI Quad Output Write Single Register



SPI Quad Output Write Multiple Register

Dual Address / Data Write

The SPI Dual Address / Data Write instruction inputs the instruction code one bit per clock and the address and data bytes two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

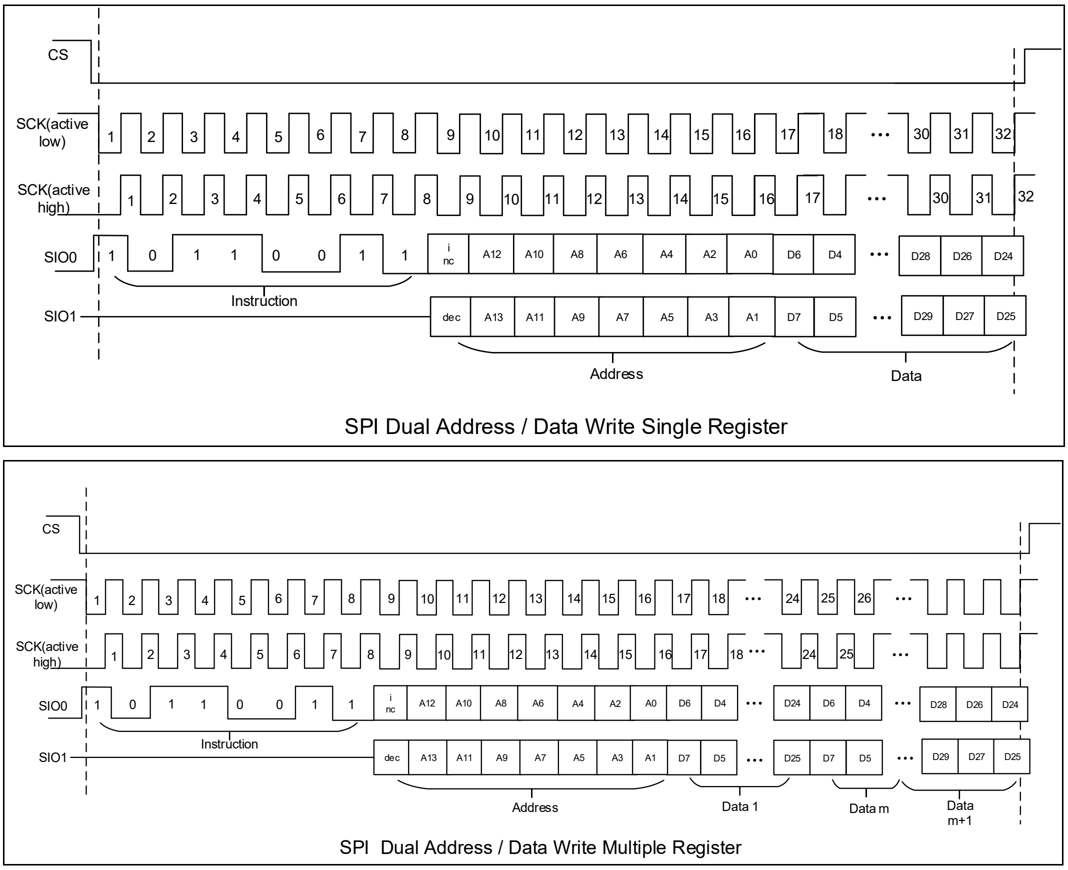
The SPI client interface is selected by first bringing CS active. The 8-bit SDADW instruction, B2h, is input into the IO[0] pin, followed by the two address bytes into the IO[1:0] pins. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the IO[1:0] pins starting with the msbs of the LSB. The remaining di-bits are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the CS is returned high, the write is considered invalid and the register is not affected.

The CS input is brought inactive to conclude the cycle.

Figure 7-20. SPI DUAL ADDRESS / DATA WRITE illustrates a typical single and multiple register dual address / data write.

Figure 7-20. SPI DUAL ADDRESS / DATA WRITE



SPI Dual Address / Data Write Single Register

SPI Dual Address / Data Write Multiple Register

Quad Address / Data Write

The SPI Quad Address / Data Write instruction inputs the instruction code one bit per clock and the address and data bytes four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in QSPI bus protocol.

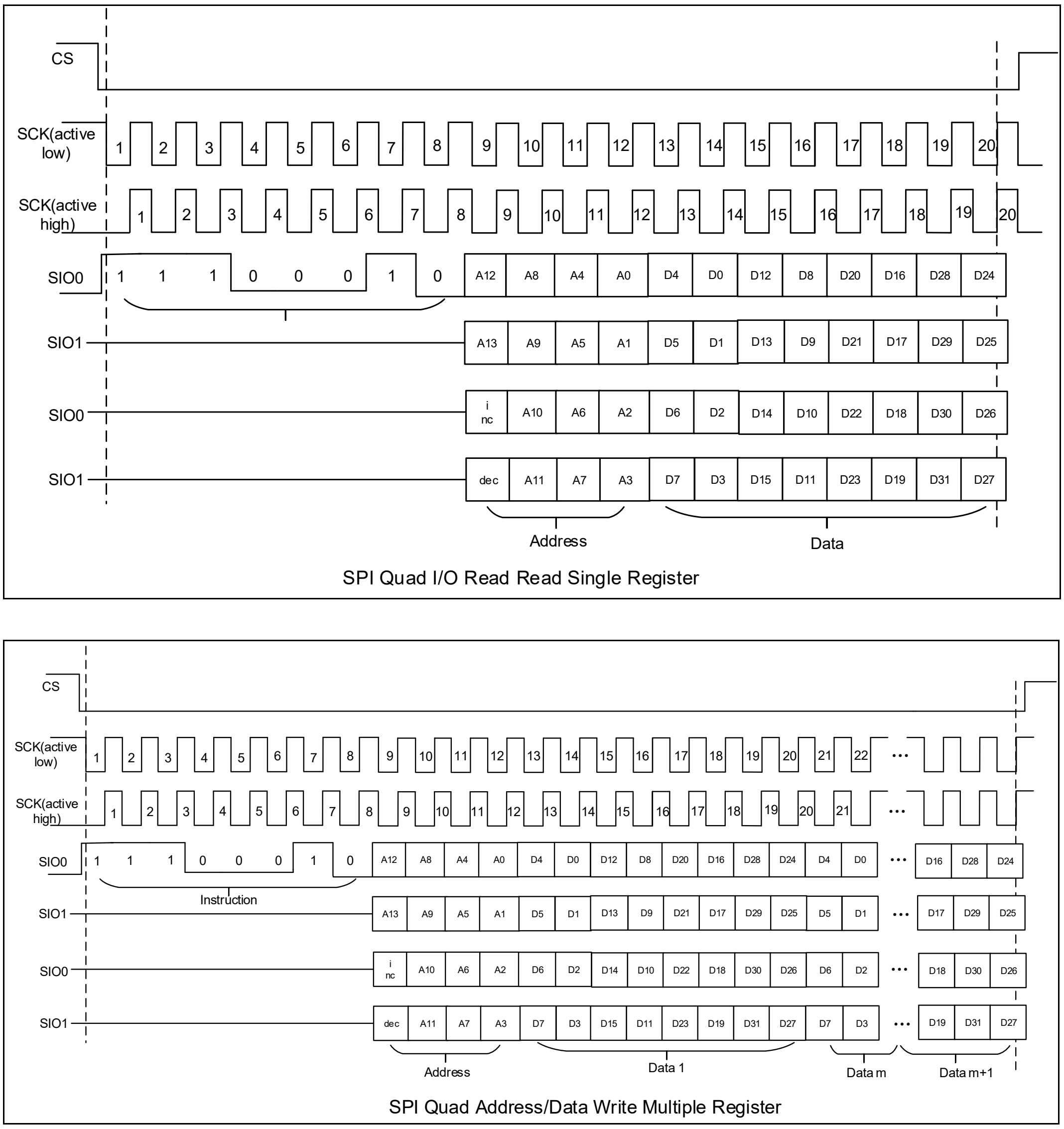
The SPI client interface is selected by first bringing CS active. The 8-bit SQADW instruction, E2h, is input into the IO[0] pin, followed by the two address bytes into the IO[3:0] pins. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the IO[3:0] pins starting with the msn of the LSB. The remaining nibbles are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the CS is returned high, the write is considered invalid and the register is not affected.

The CS input is brought inactive to conclude the cycle.

Figure 7-21. SPI QUAD ADDRESS / DATA WRITE illustrates a typical single and multiple register quad address / data write.

Figure 7-21. SPI QUAD ADDRESS / DATA WRITE


CS
SCK(active low)
SCK(active high)
SIO0
1 1 1 0 0 0 1 0
A12 A8 A4 A0 D4 D0 D12 D8 D20 D16 D28 D24
SIO1
A13 A9 A5 A1 D5 D1 D13 D9 D21 D17 D29 D25
SIO0
i nc A10 A6 A2 D6 D2 D14 D10 D22 D18 D30 D26
SIO1
dec A11 A7 A3 D7 D3 D15 D11 D23 D19 D31 D27
Address
Data
SPI Quad I/O Read Read Single Register
CS
SCK(active low)
SCK(active high)
SIO0
Instruction
SIO1
SIO0
SIO1
Address
Data 1
Data m
Data m+1
SPI Quad Address/Data Write Multiple Register

SPI WAKE UP SYSTEM

When Chip has entered Low Power mode, User can access ByteTest and READY register to exit Low Power Mode by SPI/QSPI/OSPI.

To determine when the host interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) register can be polled to determine when the device is fully awake.

SPI ACCESS FIFO

SPI/QSPI/OSPI supports access to registers and FIFO. In the process of accessing FIFO, users need to ensure that FIFO are not out of the boundary during access, otherwise data loss will occur.

The BUS module provides some FIFO Count registers, PRAM_RD_AVAIL_CNT, PRAM_WR_AVAIL_CNT, which can be read out through SPI bus. When the user reads TXFIFO, the number of read data should be less than or equal to PRAM_RD_AVAIL_CNT. When the user writes RXFIFO, the number of write data should be less than or equal to PRAM_WR_AVAIL_CNT.

When reading TXFIFO, it is recommended to read PRAM_RD_AVAIL_CNT first and then read PRAM_RD_AVAIL_CNT data in TXFIFO.

When writing RXFIFO, it is recommended to read PRAM_WR_AVAIL_CNT first, and write PRAM_WR_AVAIL_CNT data to RXFIFO.

7.2. External memory controller (EXMC)

7.2.1. Overview

The EXMC module provides an interface to connect the EXMC module in the MCU and the ESC sub controller. It can convert the signal of EXMC into the signal of AHB to realize the data transmission between MCU and ESC.

7.2.2. Characteristics

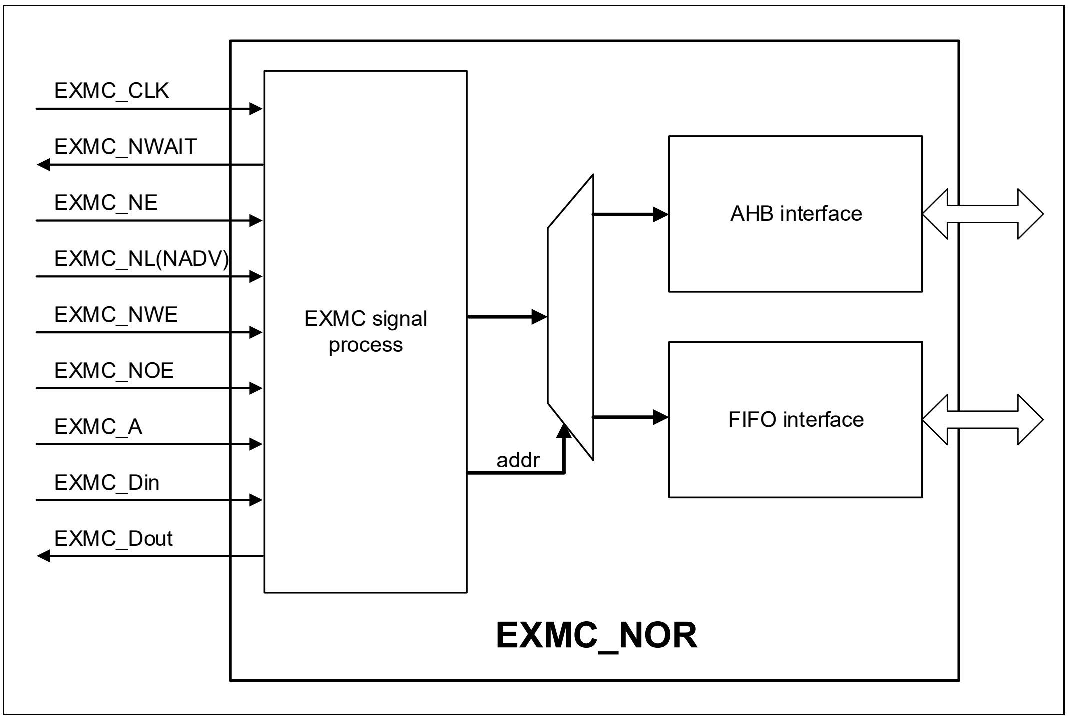
- Convert the EXMC signal to the AHB signal to access the ESC CCTL
- Convert EXMC signals into FIFO read and write signals to access ESC kernel PDRAM.
- support EXMC multiplexing mode:8-bit and 16-bit; AHB: 8-bit, 16-bit and 32-bit.
- Support host MCU clock up to 200M and sub ESC clock up to 80M.
- Support manual device wake-up via EXMC.
- Support access PDRAM

7.2.3. Function overview

Block diagram

EXMC is the combination of five modules: The AHB bus interface, EXMC configuration registers, NOR/PSRAM controller and external device interface. AHB clock (HCLK) is the reference clock.

Figure 7-22. The EXMC block diagram



Basic transmission

EXMC_NOR supports both async and sync modes. Async mode is used to access the ESC CCTL, sync mode is used to access the asynchronous FIFO. The EXMC_NOR module samples the address sent by the MCU and determines whether the current transmission is asynchronous or synchronous based on the address range.

Async mode

In async mode, all EXMC inputs change at the rising edge of the host MCU's HCLK, and ESC synchronizes these signals with the system clock. Therefore, the host needs to hold these signals for enough time to ensure that the sampling is correct.

The async mode is used to access core ESC CCTL and system ESC CCTL. Each EXMC transfer can be 8-bit or 16-bit, and supports conversion to 8-bit, 16-bit, 32-bit AHB transfer.

Table 7-7. EXMC pin and description

Pin name	Description
EXMC_NE	Chip enable. It needs to be low during EXMC transmission.
EXMC_NL(NADV)	Address enable. Indicates that the address sent is valid.
EXMC_NOE	Output enable. Indicates that data sent by ESC to MCU is valid.
EXMC_NWE	Write enable. Indicates that the data written by the MCU to ESC is valid.
EXMC_Din	Input data. Indicates the address during address sending and the data written to ESC during data write.
EXMC_Dout	Output data. Indicates the data sent by ESC to MCU

The host MCU will send address and data to ESC through EXMC. The address establishment time ASET and data establishment time DSET are four slave system clock periods, and the address hold time AHLD is one slave system clock period to ensure correct address and data sampling. An AHB transfer is initiated when the address and data are sampled. In GPIO, EXMCTYPE and EXMCHSIZE can be configured to determine the data bit width of EXMC and AHB. EXMCTYPE=1 is the 16-bit mode of EXMC, and EXMCTYPE=0 is the 8-bit mode. HSIZE is configured in the SYSCFG module. hsize=00, 01, and 10 correspond to AHB8, 16, and 32 bits respectively.

EXMC_NOR automatically determines when to convert the AHB transmission based on the bit width of EXMC and AHB, for example

When initiating a write transmission with EXMCTYPE=0 and AHB as 32-bit, EXMC_NOR will complete the AHB signal conversion after the fourth data transmission.

When a EXMCTYPE=0, AHB 32-bit read transfer is initiated, the data is read from the AHB bus when the first EXMC transfer is initiated, and 8 bits of data are transmitted to the EXMC_NOR host in each of the four transactions.

Figure 7-23. Asynchronous write transmission shows a 16-bit EXMC write operation converted to a 16-bit AHB write operation. When EXMC_NOR samples the address from the host, pull down the NWAIT signal, so that after entering the data stage, the host MCU can sample the NWAIT signal to maintain the waiting state, and when the AHB transmission ends, EXMC_NOR releases NWAIT. After NWAIT is released, the system clock release write function EXMC_NWE is enabled for three hosts, and then the system clock release chip select signal EXMC_NE is enabled.

Figure 7-23. Asynchronous write transmission

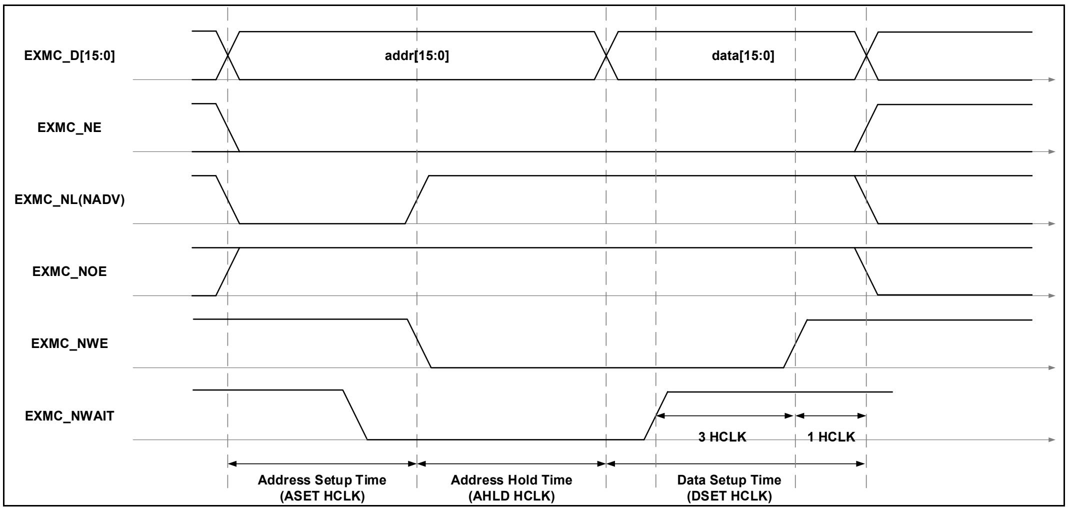


Figure 7-24. Asynchronous read transmission shows the timing diagram of asynchronous read transmission. The nwait mechanism and DSET configuration are mutually exclusive. When nwait is pulled up during the creation of read data transfer, EXMC_NOE will extend the system clock of the host by 4 hosts after nawit is pulled up, regardless of the configuration of DSET.

Figure 7-24. Asynchronous read transmission

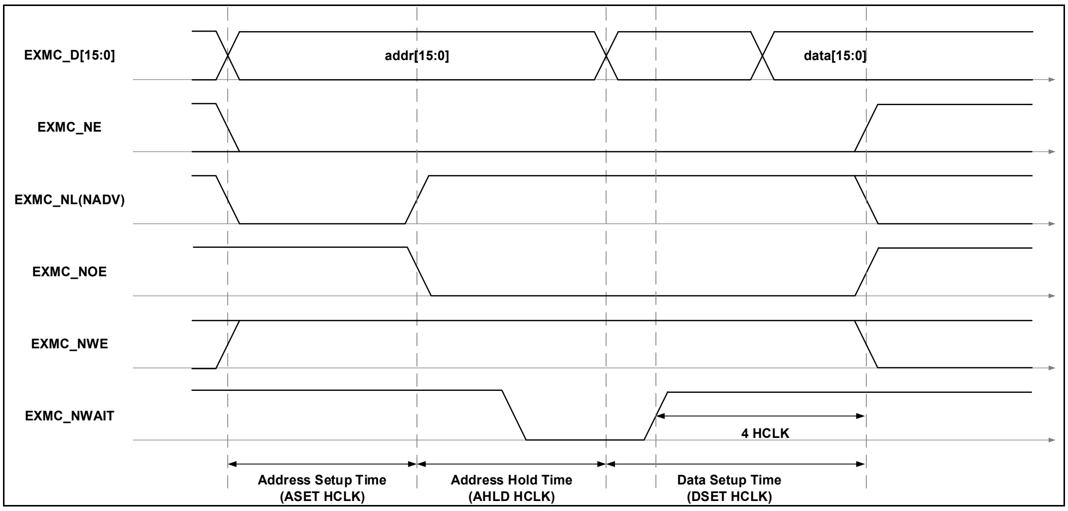
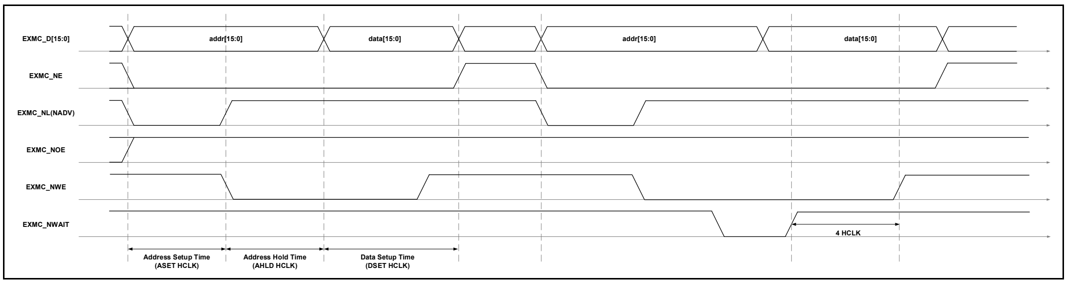


Figure 7-25. back-to-back transfers with nwait included shows back-to-back transfers with nwait included.

Figure 7-25. back-to-back transfers with nwait included



Sync mode

EXMC_NOR utilizes sync mode to access the asynchronous FIFO that sends data to the ESC core. In sync mode, all signals sent by the MCU change along the falling edge of the EXMC_CLK provided by the MCU. When the FIFO is full or read empty, the nwait signal is used to hold the MCU signal, so that the signal of nwait pulling down the beat is invalid.

Figure 7-26. Write transmission in sync mode shows the write transmission in sync mode. EXMC write data is sampled 5 cycles after the address is sampled. Every two EXMC writes are spliced into a 32bit data, and FIFO writes are enabled to complete FIFO writes.

Figure 7-26. Write transmission in sync mode

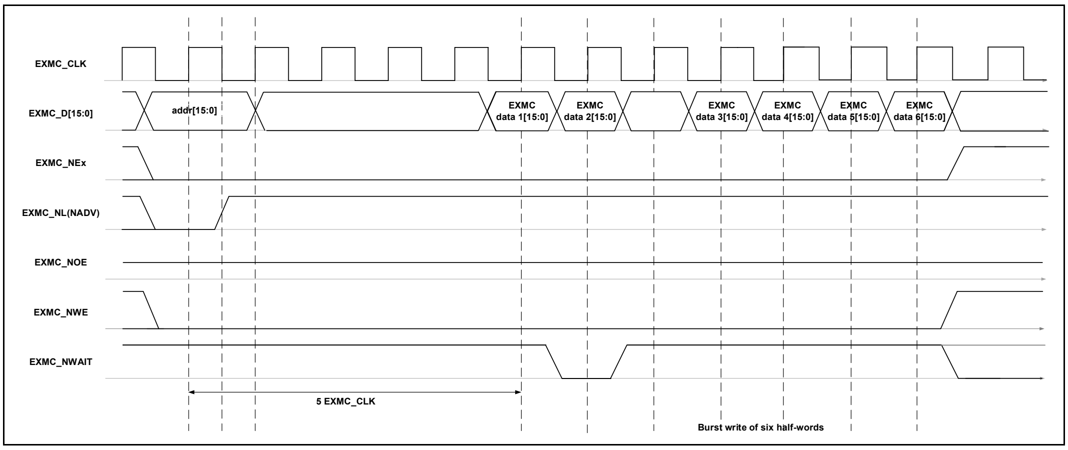
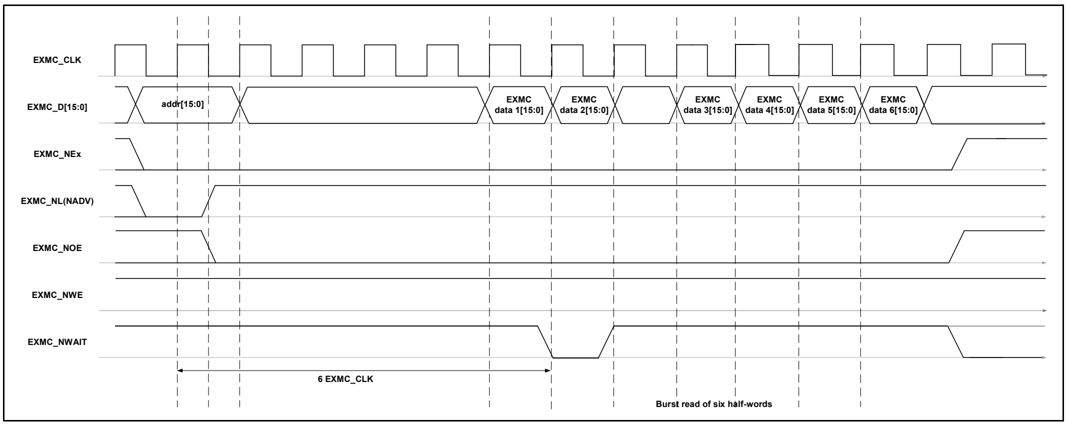


Figure 7-27. Read transmission in sync mode shows the read transmission in sync mode. When the address is sampled, the host MCU starts to sample EXMC read data six cycles later. In 16-bit mode, the read data of each FIFO is split into two 16-bit data sent successively from EXMC_NOR.

Figure 7-27. Read transmission in sync mode



Usage process

Before using EXMC_NOR, you need to poll *Process data interface reference value register (PMU_PDIREFVAL)* to check whether EXMC_NOR is available. If the unique value is 0x76543210, EXMC_NOR can be used normally. Then poll the READY bit in PMU module.

When the bit is set to 1, the entire device is ready for use.

Wake up function

When ESC enters low-power mode D1, D2H or D3, EXMC_NOR can be used to manually wake up the device by initiating a write operation to register BYTE_TEST through the EXMC_NOR interface. The AHLD of EXMC needs to be extended to 3 host MCU system clocks during this operation. After sending the wake up operation, you can determine whether EXMC_NOR and the entire device are available by polling ***Process data interface reference value register (PMU_PDIREFVAL)*** and ***Control register 0 (PMU_CTL0)*** Bit0 in ***Usage process***.

8. Ethernet PHYS

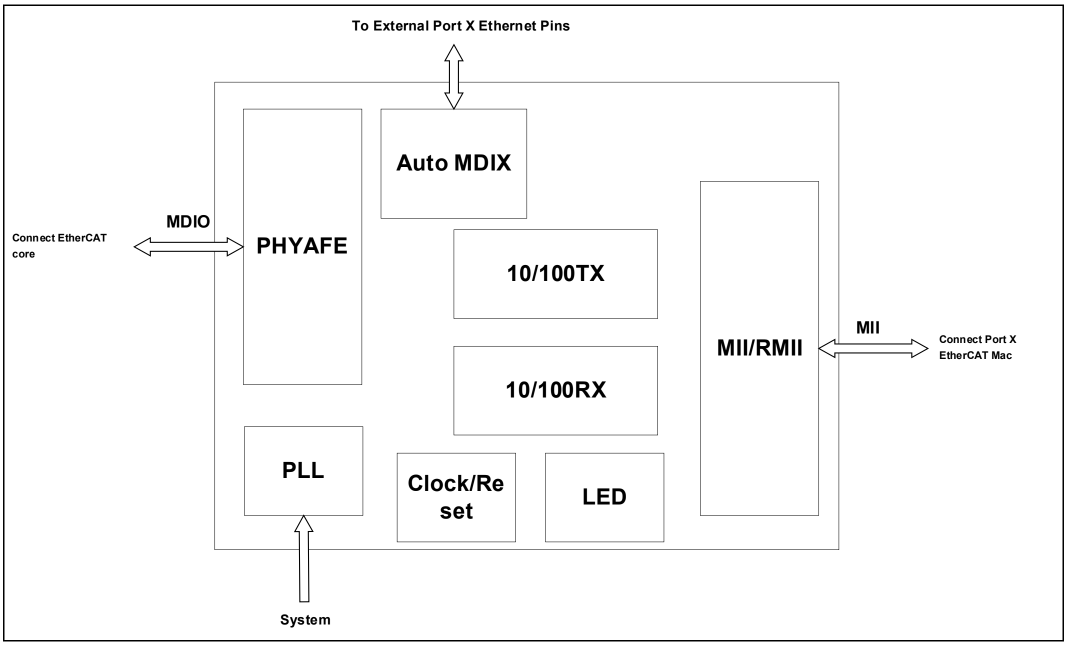
8.1. Overview

GDSCN contains PHYs A and B, there are identical in functionality. The PHY A connects to the EtherCAT port 0 or 2. The PHY B connects to EtherCAT port 1. These PHYs interface with their respective MAC via an internal MII interface. The PHYs comply with the IEEE 802.3 Physical Layer for Twisted Pair Ethernet and can be configured for full duplex 100 Mbps (100BASE-TX) Ethernet operation. All PHYs registers follow the IEEE 802.3 specified MII management register set and are fully configurable.

8.2. Characteristics

- Fully IEEE 802.3 100 Base-TX compliant and supports EEE
- Auto negotiation and parallel detection capability for automatic speed and duplex selection
- Supports MII interfaces
- Auto polarity correction in 10Base-T
- Supports Auto-MDIX function for Plug-n-Play
- Programmable loopback mode for diagnostic
- Supports programmable LED output for different applications power on LED Self-Test
- Supports WOL(Wake-On-Lan) functionality

Figure 8-1.PHY functional block diagram



8.3. Functional Overview

8.3.1. Operation Mode

100BASE-TX: In the transmitter, the data stream from the MAC interfaces is 4B/5B encoded, serialized, scrambled, and coded with a MLT3 encoder. In the receiver, the data stream from the medium is recovered, decoded from MLT3, descrambled, parallelized, and 5B/4B decoded into 4-bit data.

When there's no data to be transmitted, the system informs its link partner with Low Power Idle (LPI) signaling and then enter transmitter power-saving mode. On the other hand, when receiving the LPI signal from the link partner, the system enters receiver power-saving mode. Only periodical signaling is used to keep the link alive.

8.3.2. MII Interface

The Media Independent Interface (MII) is an interface, defined in IEEE 802.3u, between the MAC and PHY. The clock rate is equal to 2.5MHz for 10Mbps transmission and 25MHz for 100Mbps transmission. The MAC transmits and receivers data synchronously with TXCLK and RXCLK which are generated by PHY.

TXEN is asserted, TXD[3:0] is accepted for transmission by the PHY. Assertion of TXER while TXEN is asserted indicates transmit coding error. The combination of TXEN de-asserted, TXER asserted, and TXD[3:0] equal to 0001 shows a request to enter (or remain) in low power state. TXEN, TXER and TXD[3:0] are synchronously sampled with TXCLK.

When RXDV is asserted, RXD[3:0] transfer the recovered data from the PHY to MAC.

Assertion of RXER indicates a receive error. The combination of RXDV de-asserted, RXER asserted, and RXD[3:0] equal to 0001 informs it's LPI client (say MAC) that the link partner is in the low power state. CRS is asserted when the PHY is transmitting or receiving. COL is asserted when the PHY detects a collision. RXDV, RXER and RXD are synchronous with RXCLK.

8.3.3. SMI Interface

The Serial Management Interface (SMI) can be used to transfer control and status information between the Station Management (STA) and the PHY. Users can also access the internal register settings of the PHY with SMI. The MDIO is a bidirectional signal, mainly composed of command (r/w) field and data field, and synchronous with MDC. The MDIO pin should be pulled-up when there's no driving signal.

8.3.4. Automatic MDI/MDIX and Polarity Configuration

Automatic MDI/MDIX configuration is intended to eliminate the need for external crossover cables between two devices. PHY can do MDI/MDIX configuration automatically so that transmission and reception work normally. The MDI/MDIX configuration can also be determined by setting the register manually.

PHY can correct the polarity errors on the pairs of cable automatically.

8.3.5. Loopback Modes

The loopback mode provides a diagnostic function to perform the transmission and reception, so it can tests the transmit and receive data paths.

8.3.6. Wake-On-LAN

Wake-On-LAN is implemented using a special network message called a magic packet. The magic packet contains the MAC address of the destination device. The listening device waits for a legal magic packet addressed to it and then activates system wake-up procedure.

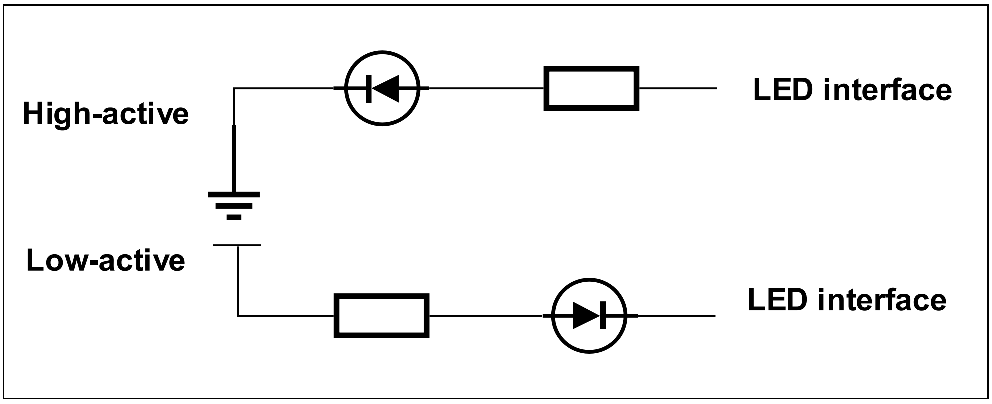
When Wake-on-Lan function is enabled, the PHY will send a interrupt after a legal magic packet is received.

8.3.7. LED Modes

There are 3 LED interface used to control LED status for link status, speed and duplex mode indicating.

There are two LED connection type: high-active and low-active, showed ***Figure 8-2. LED connect diagram*** below:

Figure 8-2. LED connect diagram



Once led interface is connected as high-active type, the interface PHY_LED_POL should be tied to zero.

Once led interface is connected as low-active type, the interface PHY_LED_POL should be tied to one.

The LED status information is defined as below:

- Link LED:
  - on: Link is up
  - off: Link is down
  - flush: Data transmission
- Speed LED:
  - on: 100M
  - off: 10M
  - flush: N/A
- Duplex LED:
  - on: Full duplex
  - off: Half duplex
  - flush: Collision

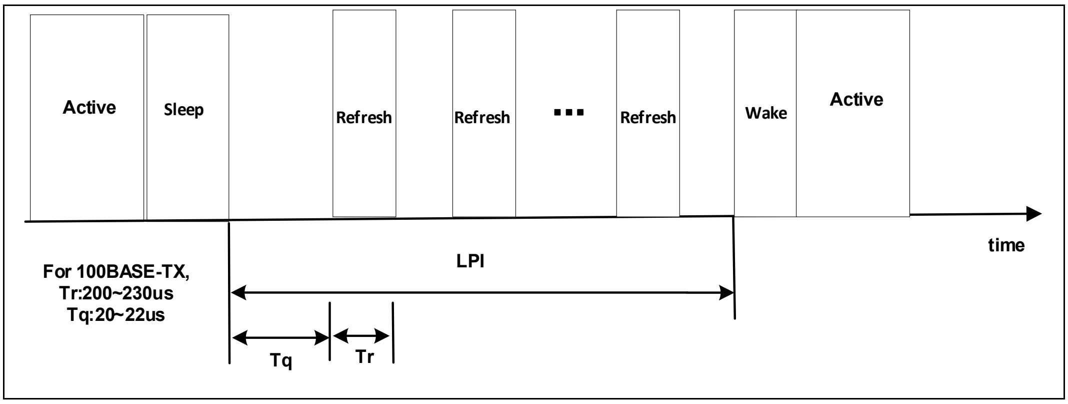
The flush period of LED is 33 milliseconds.

8.3.8. LPI Signaling

When the LPI client issues a LPI request, the PHY transmits Sleep symbols to inform its link partner that the local PHY is going to enter the LPI state. The PHY enters LPI state after transmitting Sleep symbols. During the LPI state, only Refresh symbols is transmitted periodically. When the LPI client requests to leave the LPI state, the PHY transmits Wake symbols to ask the link partner to wake-up for further transmission.

When receiving the Sleep symbols from its link partner, the PHY knows that the remote PHY is going to enter the LPI state. After the remote PHY stops transmitting, the local PHY can turn off some circuits to save power. During the LPI state, the PHY uses Refresh symbols to update its filter coefficients and adjust timing. When receiving Wake symbols from its link partner, the PHY goes back to normal operation from LPI state before a specified recovery time.

Figure 8-3. 100Base-TX LPI



8.4. PHY Register definition

8.4.1. Page 0 Registers

PHY control Register (PHY_MII_CTL)

Address offset: 0x00

Reset value: 0x3100

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR_MAIN_REST	LOOPBACK_EN	FORCE_SPEED(LSB)	MR_AUTONEG_EN	POWERDOWN	Reserved	MR_RESTART_AUTONEG	FORCE_DUPLEX	COL_TEST	FORCE_SPEED(MSB)	UNIDIRECTIONAL_EN	Reserved				
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw					

Bits	Fields	Descriptions
15	MR_MAIN_REST	Main Reset Reset the status and control register of PHY to their default values and self-clearing 1: Reset 0: Normal
14	LOOPBACK_EN	Loopback Enable When in loopback mode, the duplex will set to full duplex mode and Auto-Negotiation capability will be disabled automatically
13	FORCE_SPEED(LSB)	Force Speed LSB Bit 13 and 6 combines the speed selection. It is only valid when

		MR_AUTONEG_EN = 0. When fiber mode is enable, internal speed will be set to 100M automatically and ignores this field setting. 00: 10M 01: 100M 1X: Reserved
12	MR_AUTONEG_EN	Auto Negotiation Enable Result of this bit should be OR'ed with I_FXEN and EN_FX to determine internal final Auto-Negotiation enable signal (Auto negotiation will be disabled in Fiber mode). Please also note that when LOOPBACK_EN is set to 1
11	POWERDOWN	Power Down Mode Active high to program the PHY into power down (power down analog TX analog RX, analog AD)
10	Reserved	Must be kept at reset value.
9	MR_RESTART_AUTONEG	Re-Start Auto Negotiation This bit will be self-cleared by the PHY after programming 1'b1 to this bit. Writing 1'b0 to this bit has no effect 1: Restart Auto-Negotiation 0: Normal
8	FORCE_DUPLEX	Force Duplex Mode This bit is only valid when MR_AUTONEG_EN = 1'b0. Result of this bit should be OR'ed with LOOPBACK_EN to determine internal final duplex capability 1: Full Duplex (Default) 0: Half Duplex
7	COL_TEST	Collision Test When this bit is assert, the PHY will assert COL signal within 512 BT in response to assertion of TX_EN and will de-assert the COL signal within 4 BT when connected to MII or 16 BT when connected to GMII in response to the de-assertion of TX_EN
6	FORCE_SPEED(MSB)	Force Speed MSB Please refer to bit 13
5	UNIDIRECTIONAL_EN	Unidirectional Enable Enable the ability to encode and transmit data from the MII / GMII interface regardless of whether the PHY has determined that a valid link has been established. This ability is only valid when Auto-Negotiation is disabled and duplex mode is full
4:0	Reserved	Must be kept at reset value.

PHY status Register (PHY_MII_STATUS)

Address offset: 0x01

Reset value: 0x79C9

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
100BASE_T4	100BASE-X_FULL_DPX	100BASE-X_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	100BASE_T2_FULL_DPX	100BASE_T2_HALF_DPX	EXTENDED_STATUS	UNIDIRECTIONAL_ABLT	MF_PRB_SUP	MR_AUTONEG_CPLT	REMOTE_FAULT	AUTONEG_ABLT	LINK_STATUS	JABBER_DETECT	EXTENDED_CAP
r	r	r	r	r	r	r	r	r	r	r	r	r	rc_r	r	r

Bits	Fields	Descriptions
15	100BASE_T4	100BASE T4 Capability Not Supported. Will be 0 all the time
14	100BASE-X_FULL_DPX	100BASE TX Full Duplex Capable 1: PHY is 100BASE-X full duplex capable 0: PHY is not 100BASE-X full duplex capable
13	100BASE-X_HALF_DPX	100BASE TX Half Duplex Capable 1: PHY is 100BASE-X half duplex capable 0: PHY is not 100BASE-X half duplex capable
12	10BASE-T_FULL_DPX	10BASE-T Full Duplex Capable 1: PHY is 10BASE-T full duplex capable 0: PHY is not 10BASE-T full duplex capable
11	10BASE-T_HALF_DPX	10BASE-T Half Duplex Capable 1: PHY is 10BASE-T half duplex capable 0: PHY is not 10BASE-T half duplex capable
10	100BASE _T2_FULL_DPX	100BASE T2 Full Duplex Capable Not Supported. Will be 0 all the time
9	100BASE_T2_HALF_DPX	100BASE T2 Half Duplex Capable Not Supported. Will be 0 all the time
8	EXTENDED_STATUS	Extended Status 1: Extended status information in register 15 0: No extended status information in register 15
7	UNIDIRECTIONAL_ABLT	Unidirectional Ability 1: PHY able to transmit from MII/GMII regardless of whether the PHY has determined that a valid link has been established 0: PHY able to transmit from MII/GMII only when the PHY has determined that a valid link has been established
6	MF_PRB_SUP	Preamble Suppression Capability

		1: PHY can accept management frames with preamble suppression 0: PHY cannot accept management frames with preamble suppression
5	MR_AUTONEG_CPLT	Auto Negotiation Complete 1: Auto-negotiate completed 0: Auto-negotiate incomplete
4	REMOTE_FAULT	Remote Fault Detection 1: Remote Fault 0: Normal
3	AUTONEG_ABLT	Auto Negotiation Ability 1: PHY can Auto-Negotiate 0: PHY can not Auto-Negotiate
2	LINK_STATUS	Link Status 1: Link is up 0: Link is down Note: When the PHY is programmed into loopback mode, ***PHY control Register (PHY_MII_CTL)*** BIT14 (LOOPBACK_EN is set to 1'b1), the PHY will be forced linkup. Link Status should updated according in this bit to make MAC work properly. This bit is implemented with a latching high function, such that the occurrence of a link failure condition will cause this bit to become cleared and remain cleared until it is read.
1	JABBER_DETECT	Jabber Condition Detected 1: Jabber RX and TX condition detected 0: Normal
0	EXTENDED_CAP	Extended Register Capability 1: Extended register capabilities 0: Basic register set capabilities only

PHY Identifier Register (PHY_ID_REG)

Address offset: 0x02

Reset value: 0x0044

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID [15:0]															
r															

Bits	Fields	Descriptions
15:0	PHY_ID [15:0]	PHY ID bit [31-16] OUI (bits 3-18). OUI =00-11-05

PHY Version Register (PHY_VER_REG)

Address offset: 0x03

Reset value: 0x1400

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID[5:0]						PHY_MANF[5:0]						PHY_REV_NO[3:0]			
r						r						r			

Bits	Fields	Descriptions
15:10	PHY_ID[5:0]	PHY ID bit [5-0] OUI bits 19-24
9:4	PHY_MANF[5:0]	Manufacturer's Model Number Manufacturer's Model Number (bits 5-0) where [5:4] = architecture version
3:0	PHY_REV_NO[3:0]	Revision Number (bits3-0) Register 3, bit 0 is LS bit of PHY Identifier

Auto-Negotiation Advertisement Register (PHY_AUTONEG_ADV)

Address offset: 0x04

Reset value: 0x0101

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE_EN	Reserved	REMOTE_FAULT_EN	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASE_TX_FULL_DPX	100BASE_TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SLCT_FLD[4:0]				
rw		rw	rw	rw	rw	r	rw	rw	rw	rw	r				

Bits	Fields	Descriptions
15	NEXT_PAGE_EN	Next Page Enable 1: Set to use Next Page 0: Not to use Next Page
14	Reserved	Must be kept at reset value
13	REMOTE_FAULT_EN	Remote Fault Detection Enable 1: Auto Negotiation Fault Detected 0: No Remote Fault
12	EXTENED_NEXT_PAGE	Extended Next Page Not supported in the PHY. Should be wrote 0 all the time
11	ASYM_PAUSE	Asymmetric Pause Capability Technology Ability A6 1: Asymmetric Pause capable 0: Asymmetric Pause non-capable

10	PAUSE	Pause Capability Technology Ability A5 1: Pause capable 0: Pause non-capable
9	100BASE_T4	100BASE-T4 Capable Not supported in the PHY. Should be wrote 0 all the time
8	100BASETX_FULL_DPX	100BASE-X Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in PHY control Register (PHY_MII_CTL) BIT6, BIT13 (FORCE_SPEED) and BIT8(FORCE_DUPLEX), when FORCE_SPEED is 2'b01 and FORCE_DUPLEX is 1'b1, then this bit will be 1'b1 and vice versa
7	100BASETX_HALF_DPX	100BASE-X Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in PHY control Register (PHY_MII_CTL) BIT6, BIT13 (FORCE_SPEED) and BIT8 (FORCE_DUPLEX). when FORCE_SPEED is 2'b01 and FORCE_DUPLEX is 1'b0, then this bit will be 1'b1 and vice versa
6	10BASE-T_FULL_DPX	10BASE-T Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in PHY control Register (PHY_MII_CTL) BIT6, BIT13 (FORCE_SPEED) and BIT8 (FORCE_DUPLEX). When FORCE_SPEED is 2'b00 and FORCE_DUPLEX is 1'b1, then this bit will be 1'b1 and vice versa
5	10BASE-T_HALF_DPX	10BASE-T Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in PHY control Register (PHY_MII_CTL) BIT6, BIT13 (FORCE_SPEED) and BIT8 (FORCE_DUPLEX). when FORCE_SPEED is 2'b00 and FORCE_DUPLEX is 1'b0, then this bit will be 1'b1 and vice versa
4:0	SLCT_FLD[4:0]	Identifies Type of Message Forced to 5'h01 all the time

Auto-Negotiation Link Partner(LP) Ability Register (PHY_LP_ABILITY)

Address offset: 0x05

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	ACKNOWLEDGE	REMOTE_FAULT	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASET4	100BASETX_FULL_DPX	100BASETX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SELECTOR_FIELD[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r				

Bits	Fields	Descriptions
15	NEXT_PAGE	Link Partner Next Page Request 1: Link Partner is requesting Next Page function 0: Base Page is requested
14	ACKNOWLEDGE	Link Partner ACKNOWLEDGE Received 1: Link partner acknowledge Received Successfully 0: Not Received
13	REMOTE_FAULT	Link Partner Detects Remote Fault 1: Auto Negotiation Fault Detected 0: No Remote Fault
12	EXTENED_NEXT_PAGE	Extended Next Page
11	ASYM_PAUSE	Link Partner Asymmetric Pause Capable Technology Ability A6 1: Asymmetric Pause capable 0: Asymmetric Pause non-capable
10	PAUSE	Link Partner Symmetric Pause Capable Technology Ability A5 1: Symmetric Pause capable 0: Symmetric Pause non-capable
9	100BASET4	Technology Ability A4 Link Partner 100BASE-T4 Capable
8	100BASETX_FULL_DPX	Link Partner 100BASE-X Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
7	100BASETX_HALF_DPX	Link Partner 100BASE-X Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
6	10BASE-T_FULL_DPX	Link Partner 10BASE-T Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable

		Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
5	10BASE-T_HALF_DPX	Link Partner 10BASE-T Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
4:0	SELECTOR_FIELD[4:0]	Link Partner Identifies Type of Message Should be 5’h01

Auto-Negotiation Expansion Register (PHY_AUTONEG_EXP)

Address offset: 0x06

Reset value: 0x0064

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									NXT_PAGE_LOC_ABLE	NXT_PAGE_STO_LOC	PARA_DET_FAULT	LNK_PTNER_NP_ABLE	MR_NP_ABLE	PAGE_RECEIVED	LNK_PTNER_AN_ABLE
									r	r	r	r	r	r	r

Bits	Fields	Descriptions
15:7	Reserved	Must be kept at reset value
6	NXT_PAGE_LOC_ABLE	Received Next Page Location Able 1: Received Next Page storage location is specified by bit 5 0: Received Next Page storage location is not specified by bit 5
5	NXT_PAGE_STO_LOC	Received Next Page Storage Location 1: Link Partner next page are stored in Auto-Negotiation Next Page Received Register (PHY_AUTONEG_NEXT_PAGE_RECEIVE) 0: Link Partner next page are stored in Auto-Negotiation Link Partner(LP) Ability Register (PHY_LP_ABILITY)
4	PARA_DET_FAULT	Parallel Detect Fault 1: Local Device Parallel Detection Fault 0: No fault detected
3	LNK_PTNER_NP_ABLE	Link Partner Next Page Capable 1: Link Partner is Next Page Able 0: Link Partner is not Next Page Able
2	MR_NP_ABLE	Next Page Capable 1: Local device is Next Page Able 0: Local device is not Next Page Able

1	PAGE_RECEIVED	New Page Received 1: A New Page has been received 0: A New Page has not been received
0	LNK_PTNER_AN_ABLE	Link Partner Auto Negotiation Able 1: Link Partner is Auto-negotiation able 0: Link Partner is not Auto-negotiation able

Auto-Negotiation Next Page Transmit Register (PHY_AUTONEG_NEXT_PAGE_TRANSMIT)

Address offset: 0x07

Reset value: 0x2001

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	Reserved	MESSAGE_PAGE	ACKNOWLEDGE2	TOGGLE	MSG_UFMT_CODE_FIELD[10:0]										
rw		rw	rw	r	rw										

Bits	Fields	Descriptions
15	NEXT_PAGE	The Last Next Page Indicated whether this is the last next page 1: Additional next page will follow 0: This is the last next page
14	Reserved	Must be kept at reset value.
13	MESSAGE_PAGE	Message Page or Unformatted Page Indicated this is the message page or unformatted page 1: Message Page 0: Unformatted Page
12	ACKNOWLEDGE2	The Ability to Comply with the Message 1: Will comply with the message 0: Can not comply with the message
11	TOGGLE	Toggle The toggle bit will calculated by hardware automatically, SW can ignore
10:0	MSG_UFMT_CODE_FIELD[10:0]	Message/Unformatted Code Field

Auto-Negotiation Next Page Received Register (PHY_AUTONEG_NEXT_PAGE_RECEIVE)

Address offset: 0x08

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NP_RX[15:0]															
r															

Bits	Fields	Descriptions
15:0	NP_RX[15:0]	Next Page Received from Link Partner

MMD Access Control Register (MMD_CTL)

Address offset: 0x0D

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FUN[1:0]		Reserved									DEVAD[4:0]				
rw											rw				

Bits	Fields	Descriptions
15:14	FUN[1:0]	Function 00: address 01: data, no post increment 10: data, post increment on reads and writes 11: data, post increment on writes only
13:5	Reserved	Must be kept at reset value.
4:0	DEVAD[4:0]	Device Address

MMD Access Data Address Register (MMD_ADDR_DATA)

Address offset: 0x0E

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_DATA[15:0]															
rw															

Bits	Fields	Descriptions
15:0	ADDR_DATA[15:0]	Address Data When bit 13.15:14==0, address register Otherwise, data register

PHY Extended Status Register (PHY_EXTENDED_STATUS)

Address offset: 0x0F

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000BASE-X_FULL_DPX	1000BASE-X_HALF_DPX	1000BASE-T_HALF_DPX	1000BASE-T_HALF_DPX	Reserved											
r	r	r	r												

Bits	Fields	Descriptions
15	1000BASE-X_FULL_DPX	1000BASE TX Full Duplex Capable 1: PHY is 1000BASE-X full duplex capable 0: PHY is not 1000BASE-X full duplex capable
14	1000BASE-X_HALF_DPX	1000BASE TX Full Duplex Capable 1: PHY is 1000BASE-X full duplex capable 0: PHY is not 1000BASE-X full duplex capable
13	1000BASE-T_HALF_DPX	1000BASE-T Full Duplex Capable 1: PHY is 1000BASE-T full duplex capable 0: PHY is not 1000BASE-T full duplex capable
12	1000BASE-T_HALF_DPX	1000BASE-T Half Duplex Capable 1: PHY is 1000BASE-T half duplex capable 0: PHY is not 1000BASE-T half duplex capable
11:0	Reserved	Must be kept at reset value.

Interrupt Status Register (INT_STS)

Address offset: 0x10

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LNK_STS_CHG_INT	MGC_PKT_DET_INT	TX_LPI_RCV_INT	RX_LPI_RCV_INT	Reserved											
rc_w1	rc_w1	rc_w1	rc_w1												

Bits	Fields	Descriptions
15	LNK_STS_CHG_INT	Link Status Change INT 0: Normal 1: Link status change
14	MGC_PKT_DET_INT	Magic Packet Detect INT 0: Normal

		1: Magic packet detected
13	TX_LPI_RCV_INT	TX LPI Received INT 0: Normal 1: TX LPI received
12	RX_LPI_RCV_INT	RX LPI Received INT Mask 0: Normal 1: RX LPI received
11:0	Reserved	Must be kept at reset value.

Interrupt Mask Register (INT_MASK)

Address offset: 0x11

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LNK_STS_CHG_INT_MSK	MGC_PKT_DET_INT_MSK	TX_LPI_RCV_INT_MSK	RX_LPI_RCV_INT_MSK	Reserved											
rw	rw	rw	rw												

Bits	Fields	Descriptions
15	LNK_STS_CHG_INT_MSK	Link Status Change INT Mask When set to 1 and LINK_STS_CHG_INT bit in INT_STS register is assert to 1, the interface INT_SMI_INT_N/EXT_SMI_INT_N will be assert to 0(low active) to indicate that a magic packet detect interrupt is occured. 1: Normal 0: Interrupt is masked
14	MGC_PKT_DET_INT_MSK	Magic Packet Detect INT Mask When set to 1 and MGC_PKT_DET_INT bit in INT_STS register is assert to 1, the interface INT_SMI_INT_N/EXT_SMI_INT_N will be assert to 0(low active) to indicate that a magic packet detect interrupt is occured. 1: Normal 0: Interrupt is masked
13	TX_LPI_RCV_INT_MSK	TX LPI Received INT Mask When set to 1 and TX_LPI_RCV_INT bit in INT_STS register is assert to 1, the interface INT_SMI_INT_N/EXT_SMI_INT_N will be assert to 0(low active) to indicate that a TX LPI received interrupt is occured. 1: Normal 0: Interrupt is masked
12	RX_LPI_RCV_INT_MSK	RX LPI Received INT Mask When set to 1 and RX_LPI_RCV_INT bit in INT_STS register is assert to 1,

		the interface INT_SMI_INT_N/EXT_SMI_INT_N will be assert to 0(low active) to indicate that a RX LPI received interrupt is occured. 1: Normal 0: Interrupt is masked
11:0	Reserved	Must be kept at reset value.

Loopback Control Register (PHY_LB_CTL)

Address offset: 0x12

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_LB_XMII2MAC	Reserved	RG_LB_PCS2MAC	RG_LB_PMA2MAC	RG_LB_AFE2MAC	RG_LB_EPG2EPC	Reserved		RG_LB_MMII2PHY	Reserved						
rw	rw	rw	rw	rw	rw			rw							

Bits	Fields	Descriptions
15	RG_LB_XMII2MAC	XMII2MAC Loopback Enable
14	Reserved	Must be kept at reset value.
13	RG_LB_PCS2MAC	PCS2MAC Loopback Enable
12	RG_LB_PMA2MAC	PMA2MAC Loopback Enable
11	RG_LB_AFE2MAC	AFE2MAC Loopback Enable
10	RG_LB_EPG2EPC	EPG2EPC Loopback Enable
9:8	Reserved	Must be kept at reset value.
7	RG_LB_MMII2PHY	RMII2PHY Loopback Enable Only valid when internal EPHY used.
6:0	Reserved	Must be kept at reset value.

PHY Global Configuration Register (PHY_GLOBAL_CONFIG)

Address offset: 0x13

Reset value: 0x0102

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		XMII_TXC_INV	XMII_RXC_INV	EN_FX	EN_WOL	FORCE_LINK	RG_WOL_RCV_BC	RG_WOL_CHK_PSWD	Reserved					MDI_MD[1:0]	
		rw	rw	rw	rw	rw	rw	rw						rw	

Bits	Fields	Descriptions
15:14	Reserved	Must be kept at reset value.
13	XMII_TXC_INV	XMII TXCLK Inversed 0: No inverse about TXCLK on the XMII interface 1: Inverse TXCLK on the XMII interface
12	XMII_RXC_INV	XMII RXCLK Inversed 0: No inverse about RXCLK on the XMII interface 1: Inverse RXCLK on the XMII interface
11	EN_FX	Fiber Enable This bit will be OR'ed with EPHY_FXEN to determine the medium type of EPHY. Below shows the definition for OR'ed result 1: Fiber Mode 0: Twisted Pair Mode Note: When EPHY is programmed into Fiber mode, Auto MDIX should be disabled automatically and force to MDI mode
10	EN_WOL	Wake-On-Lan Enable Enable Magic Packet Detect Function
9	FORCE_LINK	Force Link Up 1: Force both 10/100M Module link up 0: Force Link or not depends upon FORCE_LINK_10 and/or FORCE_LINK_100 in 10/100M Configuration Register, respectively Note: FORCE LINK works only when PHY is programmed into Force 10 or 100M mode. When PHY is programmed into auto negotiation mode, program 1'b1 to this bit has no effect (If this bit is programmed to 1'b1 ahead of auto negotiation is enabled, it should be disabled automatically by H/W
8	RG_WOL_RCV_BC	Enable Receive Broadcast Packet in WOL Mode 1: Enable receive broadcast magic packet 0: Otherwise
7	RG_WOL_CHK_PSWD	Enable SecureOn Password Check in WOL Mode 1: Enable SecureOn password check 0: Otherwise
6:2	Reserved	Must be kept at reset value.
1:0	MDI_MD[1:0]	MDI/MDIX Mode Values on DUPCOLLED and RXER will be latched during power on reset and stored in these two bits 00: Force MDI Mode 01: Force MDIX Mode 10: Auto MDI/MDIX Detection (Default) 11: Reserved MDI/MDIX mode will be set to 'Force MDI Mode' automatically when PHY is in

either Loopback or Fiber mode

MAC Address Register 0 (RG_MAC_AADR_0)

Address offset: 0x16

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_ADDR_BYTE0[7:0]								MAC_ADDR_BYTE1[7:0]							
rw								rw							

Bits	Fields	Descriptions
15:8	MAC_ADDR_BYTE0[7:0]	MAC Address Byte 0 in Transmission Order
7:0	MAC_ADDR_BYTE1[7:0]	MAC Address Byte 1 in Transmission Order

MAC Address Register 1 (RG_MAC_AADR_1)

Address offset: 0x17

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_ADDR_BYTE2[7:0]								MAC_ADDR_BYTE3[7:0]							
rw								rw							

Bits	Fields	Descriptions
15:8	MAC_ADDR_BYTE2[7:0]	MAC Address Byte 2 in Transmission Order
7:0	MAC_ADDR_BYTE3[7:0]	MAC Address Byte 3 in Transmission Order

MAC Address Register 2 (RG_MAC_AADR_2)

Address offset: 0x18

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAC_ADDR_BYTE4[7:0]								MAC_ADDR_BYTE5[7:0]							
rw								rw							

Bits	Fields	Descriptions
15:8	MAC_ADDR_BYTE4[7:0]	MAC Address Byte 4 in Transmission Order
7:0	MAC_ADDR_BYTE5[7:0]	MAC Address Byte 5 in Transmission Order

PHY Status Register (PHY_STATUS)

Address offset: 0x19

Reset value: 0x8800

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINE_NOSIG	Reserved		MR_AUTONEG_CPLT	A_TXSAVE80	A_TXSAVE50	JABTX_STATUS	MDIX_STATUS	mac_addr_byte5	POLARITY_REV	FINAL_PAUSE_OUT	FINAL_PAUSE_IN	FINAL_DUPLEX	FINAL_SPEED[1:0]		FINAL_LINK
r			r	r	r	r	r	r	r	r	r	r	r		r

Bits	Fields	Descriptions
15	LINE_NOSIG	No Signal on the Medium 0: Signal detected in the medium 1: No signal detected in the medium
14:13	Reserved	Must be kept at reset value.
12	MR_AUTONEG_CPLT	Auto Negotiation Complete 1: Complete 0: Not Complete
11	A_TXSAVE80	10M Transmit 80% Amplitude Status 1: 80% TX Amplitude 0: Normal Amplitude
10	A_TXSAVE50	10M Transmit 50% Amplitude Status 1: 50% TX Amplitude 0: Normal Amplitude
9	JABRX_STATUS	Real Time RX Jabber Status 1: RX Jabber 0: No RX Jabber
8	JABTX_STATUS	Real Time TX Jabber Status 1: TX Jabber 0: No TX Jabber
7	MDIX_STATUS	MDIX STATUS 1: MDIX 0: MDI
6	POLARITY_REV	Polarity Status 1: Reversed 0: Normal
5	FINAL_PAUSE_OUT	Pause Out Capability When auto negotiation is enabled, this bit is determined by ***Auto-Negotiation***

		Advertisement Register (PHY AUTONEG ADV) BIT [11:10] and Auto-Negotiation Link Partner(LP) Ability Register (PHY LP ABILITY) BIT[11:10] after link up. When auto negotiation is disabled, this bit will be set to 1'b0 all the time 1: With Pause Out capability 0: Without Pause Out capability
4	FINAL_PAUSE_IN	Pause In Capability When auto negotiation is enabled, this bit is determined by Auto-Negotiation Advertisement Register (PHY AUTONEG ADV) BIT [11:10] and Auto-Negotiation Link Partner(LP) Ability Register (PHY LP ABILITY) BIT[11:10] after link up. When auto negotiation is disabled, this bit will be set to 1'b0 all the time 1: With Pause In capability 0: Without Pause In capability
3	FINAL_DUPLEX	Duplex Status Before link up, S/W should ignore the status of this due to meaningless 1: Full Duplex 0: Half Duplex
2:1	FINAL_SPEED[1:0]	Speed Status Before link up, S/W should ignore the status of this due to meaningless 1: 100M 0: 10M
0	FINAL_LINK	Link Status 1: Link Up 0: Link Down

Wake-On-Lan SecureOn Password Register 0 (RG_WOL_PASSWORD_0)

Address offset: 0x1A

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_WOL_PASSWORD_BYTE0[7:0]								RG_WOL_PASSWORD_BYTE1[7:0]							
rw								rw							

Bits	Fields	Descriptions
15:8	RG_WOL_PASSWORD_BYTE0[7:0]	SecureON Password Byte 0 in Transmission Order
7:0	RG_WOL_PASSWORD_BYTE1[7:0]	SecureON Password Byte 1 in Transmission Order

Wake-On-Lan SecureOn Password Register 1 (RG_WOL_PASSWORD_1)

Address offset: 0x1B

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_WOL_PASSWORD_BYTE2[7:0]								RG_WOL_PASSWORD_BYTE3[7:0]							
rw								rw							

Bits	Fields	Descriptions
15:8	RG_WOL_PASSWORD_BYTE2[7:0]	SecureON Password Byte 2 in Transmission Order
7:0	RG_WOL_PASSWORD_BYTE3[7:0]	SecureON Password Byte 3 in Transmission Order

Wake-On-Lan SecureOn Password Register 2 (RG_WOL_PASSWORD_2)

Address offset: 0x1C

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_WOL_PASSWORD_BYTE4[7:0]								RG_WOL_PASSWORD_BYTE5[7:0]							
rw								rw							

Bits	Fields	Descriptions
15:8	RG_WOL_PASSWORD_BYTE4[7:0]	SecureON Password Byte 4 in Transmission Order
7:0	RG_WOL_PASSWORD_BYTE5[7:0]	SecureON Password Byte 5 in Transmission Order

Page Selection Register (PHY_PAGE_SEL)

Address offset: 0x1F

Reset value: 0x003D

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PAGE_SEL[4:0]					Reserved							
			rw												

Bits	Fields	Descriptions
15:13	Reserved	Must be kept at reset value.
12:8	PAGE_SEL[4:0]	MII Register Page Selection
7:0	Reserved	Must be kept at reset value.

8.4.2. Page 1 Registers

EEE Configure Register (EEE_CFG)

Address offset: 0x17

Reset value: 0x0033

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												IEEE_ENABLE	Reserved		
												rw			

Bits	Fields	Descriptions
15:4	Reserved	Must be kept at reset value.
3	IEEE_ENABLE	Intelligent EEE Enable Enable EPHY TX enter LPI state automatically when no data transmission
2:0	Reserved	Must be kept at reset value.

8.4.3. Page 2 Registers

10M Power Save Control Register (PHY_10M_PWRSAVE)

Address offset: 0x17

Reset value: 0x04C8

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYPASS_TXSAVE80	BYPASS_TXSAVE50	TEST_TXSAVE80	TEST_TXSAVE50	LINKPULSE_DLY_TH[3:0]				SAVE_80_PERCENT	TX10SAVE_MODE	Reserved	SAVE_ON_DLY_TH[4:0]				
rw	rw	rw	rw	rw				rw	rw		rw				

Bits	Fields	Descriptions
15	BYPASS_TXSAVE80	Bypass Normal Power Saving 80 Percent Path
14	BYPASS_TXSAVE50	Bypass Normal Power Saving 50 Percent Path
13	TEST_TXSAVE80	Power saving 80 percent Test Input Valid only when BYPASS_TXSAVE80 is enabled
12	TEST_TXSAVE50	Power saving 50 percent Test Input Valid only when BYPASS_TXSAVE50 is enabled
11:8	LINKPULSE_DLY_TH[3:0]	Link Pulse Delay Generate Threshold The threshold to delay transmission of TX link pulse. The time delayed is the value programmed x 2 cycles, where 1 cycle is equal to 40ns. The default value set here is to delay 320ns
7	SAVE_80_PERCENT	Power Saving 80 Percent 1: Enable

		0: Disable
6	TX10SAVE_MODE	10M Power Saving Mode 1: 10M Power Saving mode enable 0: Normal mode without power saving
5	Reserved	Must be kept at reset value.
4:0	SAVE_ON_DLY_TH[4:0]	Power Saving on Delay Threshold

Analog Transmit Data Test and Control Register (PHY_TXDATA_CTRL)

Address offset: 0x18

Reset value: 0x1000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	FIR_10_SEL[2:0]			Reserved		BYPASS_TX_10_DATA	BYPASS_TX_10_DATA	TEST_TX_100_DATA [1:0]		Reserved	TEST_TX_10_DATA [4:0]				
	rw					rw	rw	rw			rw				

Bits	Fields	Descriptions
15	Reserved	Must be kept at reset value.
14:12	FIR_10_SEL[2:0]	10M TX Filter Selection
11:10	Reserved	Must be kept at reset value.
9	BYPASS_TX_100_DATA	Bypass 100M Transmit data 1: bypass normal 100M data to analog and force TEST_TX_100_DATA to analog 0: Normal 100M data sent to analog block
8	BYPASS_TX_10_DATA	Bypass 10M Transmit data 1: bypass normal 10M data to analog and force TEST_TX_10_DATA to analog 0: Normal 10M data sent to analog block
7:6	TEST_TX_100_DATA[1:0]	100M Test Data Fed into analog block when BYPASS_TX_100_DATA is set to 1
5	Reserved	Must be kept at reset value.
4:0	TEST_TX_10_DATA[4:0]	10M Test Data Fed into analog block when BYPASS_TX_10_DATA is set to 1

8.4.4. Page 3 Registers

DSPSM Control Register (PHY_DSPSM_CTRL)

Address offset: 0x11

Reset value: 0x8510

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			BYPASS_AGCSELD	Reserved	INI_100_AGCSEL[2:0]			Reserved				BYPASS_BOOSTEND	TEST_BOOSTEND	Reserved	
			rw			rw						rw	rw		

Bits	Fields	Descriptions
15:13	Reserved	Must be kept at reset value.
12	BYPASS_AGCSELD	Bypass AGCSEL 1: Bypass normal agcselD 0: Normal
11	Reserved	Must be kept at reset value.
10:8	INI_100_AGCSEL[2:0]	100M Mode AGCSEL initial Value Function of these 3 bits are different according to the setting of BYPASS_AGCSELD BYPASS AGCSELD = 1: Internal AGCSEL will fixed to INI_100_AGCSEL BYPASS AGCSELD = 0 : INI_100_AGCSEL will be loaded during DSPRST and internal boosten is 1’b0, and will be kept without change when internal boosten is low. Therefore, DSPSM can cover [boosten, AGCSEL] for the range of [0,000] and [1,000] to [1,111]
7:4	Reserved	Must be kept at reset value.
3	BYPASS_BOOSTEND	BoostenD BypassMode Internal BoostenD will be skipped and TEST_BOOSTEND will be used to control for debugging purpose 1: Bypass mode 0: Normal mode
2	TEST_BOOSTEND	BoostenD Test Input Force boostenD to this value when BYPASS_BOOSTEND = 1’b1
1:0	Reserved	Must be kept at reset value.

8.4.5. Page 6 Registers

Analog ADC Control Register (PHY_ADC_CTL)

Address offset: 0x10

Reset value: 0x5563

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								A_CLKTX RXSEL	Reserved						
								rw							

Bits	Fields	Descriptions
15:8	Reserved	Must be kept at reset value.
7	A_CLKTXRXSEL	Bypass RXCLK125 to TXCLK125 1: Internal RXCLK125 selects TXCLK125 0: Internal RXCLK125 selects RXCLK125 (Default)
6:0	Reserved	Must be kept at reset value.

Analog Pre-Gain and PLL Configuration Register (PHY_PGPLL_CTL)

Address offset: 0x12

Reset value: 0x0D00

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SELCKAD T	Reserved													
	rw														

Bits	Fields	Descriptions
15	Reserved	Must be kept at reset value.
14	SELCKADT	Test Mode ADC Clock Select This bit is only available when PHY is in AFE test mode, other than AFE test mode, writing value to this bit has no effect 1: Select CKADTEST as ADC input clock 0: Select RXCLK125 as ADC input clock
13:0	Reserved	Must be kept at reset value.

8.4.6. Page 9 Registers

Embedded Packet Generator and Checker Command Register (EPGC_CMD)

Address offset: 0x10

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_EPG_EN	RG_EPG_DATA_TYPE[2:0]			RG_EPG_IPG_LEN_FIXED	RG_EPG_DASA_FIXED	RG_EPG_MODE[1:0]		RG_EPG_LPI_INDICATION	RG_EPC_LOC	Reserved		RG_EPC_CLR_CNT	RG_EPG_CLR_CNT	RG_EPG_PAUSE	RG_EPG_GO
rw	rw	rw	rw	rw	rw	rw		rw	rw			rw	rw	rw	rw

Bits	Fields	Descriptions
15	RG_EPG_EN	Embedded Packet Generator Enable This bit is used to enable embedded packet generator for debugging purpose. When it is asserted, internal tx data path will be switched to the embedded packet generator

14:12	RG_EPG_DATA_TYPE[2:0]	Embedded Packet Generator Mode
		3’b000: all zeros;
		3’b001: all ones;
		3’b010: all 5s;
		3’b011: all As;
		3’b100: byte Increment;
		3’b101: random;
		3’b110: byte decrement

11	RG_EPG_IPG_LEN_FIXED	Inter-Packet-Gap Length Fixed 1: Inter-Packet-Gap length fixed and determined by RG_EPG_IPG_LEN 0: Inter-Packet-Gap length randomized by hardware
10	RG_EPG_DASA_FIXED	DA/SA Fixed 1: DA fixed to 00-01-02-03-04-05 and SA fixed to 0a-0b-0c-0d-0e-0f 0: DA/SA is configured by RG_EPG_DATA_TYPE

9:8	RG_EPG_MODE[1:0]	Embedded Packet Generator Mode
		2’b00: single mode;
		2’b01: burst mode;
		2’b1x: continue Mode

7	RG_EPG_LPI_INDICATION	TX LPI indication 1: Transmit LPI 0: Normal
6	RG_EPC_LOC	EPC Location

		1: EPC is located at TX path 0: EPC is located at RX path
5:4	Reserved	Must be kept at reset value.
3	RG_EPC_CLR_CNT	Embedded Packet Checker Counter Clear High active to clear rx packet checker statistic counter, include total packet number counter and crc error counter, it is self-cleared
2	RG_EPG_CLR_CNT	Embedded Packet Generator Counter Clear High active to clear tx packet generator statistic counter, include total packet generated counter, it is self-cleared
1	RG_EPG_PAUSE	Embedded Packet Generator Packet Generation Pause Combined with RG_EPG_GO to control packet generator, { RG_EPG_PAUSE, RG_EPG_GO } = 2'b01: start; 2'b11: pause; 2'b00: stop Note: In single mode(RG_EPG_MODE[1:0]=2'b0), only the start command is valid. The other two commands will not work In burst mode(RG_EPG_MODE[1:0]=2'b01), all the three commands are valid, the pause command will pause packet generation, internal burst packet number counter will hold, and followed start command will continue the current burst generation. In continue mode(RG_EPG_MODE[1:0]=2'b10), all the three commands are valid, and the pause command behavior will be the same with the stop command.
0	RG_EPG_GO	Embedded Packet Generator Packet Generation Go Combined with RG_EPG_PAUSE to control packet generator, please refer to RG_EPG_PAUSE for the control command definition. Note: When in single mode and continue mode, this bit will be self-cleared when generation task is finished. When in continue mode, only write zero to this bit can clear it.

Embedded Packet Generator Packet Length (EPG_PKT_LEN)

Address offset: 0x11

Reset value: 0x0000

This register can be accessed by byte(8-bit), half-word(16-bit) and word(32-bit)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RG_EPG_PKT_LEN_FIXED	RG_EPG_PKT_LEN[10:0]										
				rw	rw										

Bits	Fields	Descriptions
15:12	Reserved	Must be kept at reset value.
11	RG_EPG_PKT_LEN_FIXED	Packet Length Fixed Enable 1: Packet length fixed and determined by rg_epg_pkt_len 0: Packet length randomized by hardware
10:0	RG_EPG_PKT_LEN[10:0]	Packet Length Packet total length, include the DA/SA, data and FCS. It is only valid when rg_epg_pkt_len_fixed is 1'b1, counted by byte, otherwise, packet length will be randomized by hardware

Embedded Packet Generator Inter-Packet-Gap (epg_ipg_cfg)

Address offset: 0x12

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_EPG_IPG_LEN [15:0]															
rw															

Bits	Fields	Descriptions
15:0	RG_EPG_IPG_LEN [15:0]	Inter-Packet-Gap Length

Embedded Packet Generator Burst Number High Data (EPG_BURST_PKT_NUM_CFG_HIGH)

Address offset: 0x13

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_EPG_BURST_PKT_NUMBER_HIGH [15:0]															
rw															

Bits	Fields	Descriptions
15:0	RG_EPG_BURST_PKT_NUMBER_HIGH [15:0]	Packet Number of a Burst High Data Only valid in burst mode (RG_EPG_MODE==2'b10)

Embedded Packet Generator Burst Number Low Data (EPG_BURST_PKT_NUM_CFG_LOW)

Address offset: 0x14

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RG_EPG_BURST_PKT_NUMBER_LOW [15:0]															
rw															

Bits	Fields	Descriptions
15:0	RG_EPG_BURST_PKT_NUMBER_LOW [15:0]	Packet Number of a Burst Low Data Only valid in burst mode (RG_EPG_MODE==2'b10)

TX Byte Counter High Data (TX_BYTE_CNT_HIGH)

Address offset: 0x15

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_BYTE_CNT_HIGH[15:0]															
r															

Bits	Fields	Descriptions
15:0	TX_BYTE_CNT_HIGH[15:0]	Embedded Packet Generator Packet Generation Byte Counter High Data(bit 31:16) This counter will be cleared by asserting RG_EPG_CLR_CNT

TX Byte Counter Low Data (TX_BYTE_CNT_LOW)

Address offset: 0x16

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_BYTE_CNT_LOW[15:0]															
r															

Bits	Fields	Descriptions
15:0	TX_BYTE_CNT_LOW[15:0]	Embedded Packet Generator Packet Generation Byte Counter High Data(bit 15:0) This counter will be cleared by asserting RG_EPG_CLR_CNT

TX Total Packet Counter High Data (TX_PKT_CNT_HIGH)

Address offset: 0x17

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

TX_PKT_CNT_HIGH[15:0]
r

Bits	Fields	Descriptions
15:0	TX_PKT_CNT_HIGH[15:0]	Embedded Packet Generator Total Packet Generation Counter High Data(bit 31:16) This counter will be cleared by asserting RG_EPG_CLR_CNT

TX Total Packet Counter Low Data (TX_PKT_CNT_LOW)

Address offset: 0x18

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_PKT_CNT_LOW[15:0]															
r															

Bits	Fields	Descriptions
15:0	TX_PKT_CNT_LOW[15:0]	Embedded Packet Generator Total Packet Generation Counter Low Data(bit 15:0) This counter will be cleared by asserting RG_EPG_CLR_CNT

RX Byte Counter High Data (RX_BYTE_CNT_HIGH)

Address offset: 0x19

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_BYTE_CNT_HIGH[15:0]															
r															

Bits	Fields	Descriptions
15:0	RX_BYTE_CNT_HIGH[15:0]	Packet Received Byte Counter High Data(bit 31:16) This counter will be cleared by asserting RG_EPC_CLR_CNT

RX Byte Packet Counter Low Data (RX_BYTE_CNT_LOW)

Address offset: 0x1A

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_BYTE_CNT_LOW[15:0]															
r															

Bits	Fields	Descriptions
15:0	RX_BYTE_CNT_LOW[15:0]	Packet Received Byte Counter Low Data(bit 15:0) This counter will be cleared by asserting RG_EPC_CLR_CNT

RX Total Packet Counter High Data (RX_PKT_CNT_HIGH)

Address offset: 0x1B

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_PKT_CNT_HIGH[15:0]															
r															

Bits	Fields	Descriptions
15:0	RX_PKT_CNT_HIGH[15:0]	Total Packet Received Counter High Data(bit 31:16) This counter will be cleared by asserting RG_EPC_CLR_CNT

RX Total Packet Counter Low Data (RX_PKT_CNT_LOW)

Address offset: 0x1C

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_PKT_CNT_LOW[15:0]															
r															

Bits	Fields	Descriptions
15:0	RX_PKT_CNT_LOW[15:0]	Total Packet Received Counter Low Data(bit 15:0) This counter will be cleared by asserting RG_EPC_CLR_CNT

RX CRC Error Packet Counter High Data (RX_CRC_ERR_CNT_HIGH)

Address offset: 0x1D

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CRC_ERR_CNT_HIGH [15:0]															
r															

Bits	Fields	Descriptions
15:0	RX_CRC_ERR_CNT_HIGH[15:0]	CRC Error Packet Received Counter High Data(bit 31:16)

This counter will be cleared by asserting RG_EPC_CLR_CNT

RX CRC Error Packet Counter Low Data (RX_CRC_ERR_CNT_LOW)

Address offset: 0x1E

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RX_PKT_CNT_LOW[15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bits	Fields	Descriptions
15:0	RX_CRC_ERR_CNT_LOW[15:0]	CRC Error Packet Received Counter Low Data(bit 15:0) This counter will be cleared by asserting RG_EPC_CLR_CNT

8.4.7. MDIO Registers

PCS Control 1 Register (PCS_CTL_1)

Device Address: 0x3

Address offset: 0x00

Reset value: 0x0400

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>PCS_RST</td><td colspan="4">Reserved</td><td>CLOCK_STOP_EN</td><td colspan="10">Reserved</td></tr>
<tr><td>rw</td><td colspan="4"></td><td>rw</td><td colspan="10"></td></tr>
</table>

Bits	Fields	Descriptions
15	PCS_RST	PCS Reset Reset the AN and PCS MMD registers to their default value and also cause a software reset in the PHY, it will be selfclearing after reset is finished
14:11	Reserved	Must be kept at reset value.
10	CLOCK_STOP_EN	xMII Receive Clock Stop Enable Set to 1 to stop the receive xMII clock while it is signaling LPI otherwise it will keep the clock active
9:0	Reserved	Must be kept at reset value.

PCS Status 1 Register (PCS_STS_1)

Device Address: 0x3

Address offset: 0x01

Reset value: 0x0040

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TX_LPI_REC	RX_LPI_REC	TX_LPI_INT	RX_LPI_INT	Reserved	CLOCK_STOP_CAP	Reserved					
				r	r	r	r		r						

Bits	Fields	Descriptions
15:12	Reserved	Must be kept at reset value.
11	TX_LPI_REC	TX PCS has Received LPI It is implemented with a latching high function, such that the TX LPI signaling causes this bit to become one and remain one until it is read. 1: TX PCS has received LPI 0: LPI not received
10	RX_LPI_REC	RX PCS has Received LPI It is implemented with a latching high function, such that the RX LPI signaling causes this bit to become one and remain one until it is read. 1: RX PCS has received LPI 0: LPI not received
9	TX_LPI_INT	TX PCS Receiving LPI Indication 1: TX PCS is currently receiving LPI 0: PCS is not currently receiving LPI
8	RX_LPI_INT	RX PCS Receiving LPI Indication 1: RX PCS is currently receiving LPI 0: PCS is not currently receiving LPI
7	Reserved	Must be kept at reset value.
6	CLOCK_STOP_CAP	Transmit xMII Clock Stop Capable 1: RS may stop the transmit xMII clock during LPI 0: Transmit xMII clock not stoppable
5:0	Reserved	Must be kept at reset value.

PCS Device Identifier (PCS_ID)

Device Address: 0x3

Address offset: 0x02

Reset value: 0x0044

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID[15:0]															

r

Bits	Fields	Descriptions
15:0	PHY_ID[15:0]	PHY ID bit[31-16] OUI (bits 3-18) .OUI =00-11-05

PCS Device Version Register (PCS_VER)

Device Address: 0x3
Address offset: 0x03
Reset value: 0x1400

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID[5:0]						PHY_MAN_MODEL[5:0]						PHY_REV_NO[3:0]			
r						r						r			

Bits	Fields	Descriptions
15:10	PHY_ID[5:0]	PHY ID bit[15-10] OUI bits 19-24
9:4	PHY_MAN_MODEL[5:0]	Manufacturer's Model Number Manufacturer's Model Number (bits 5-0) where [5:4] = architecture version
3:0	PHY_REV_NO[3:0]	Revision Number (bits3-0) *PCS Device Identifier (PCS_ID)* bit 0 is LS bit of PHY Identifier

PCS Package Register 0 (PCS_PKG_0)

Device Address: 0x3
Address offset: 0x05
Reset value: 0x0089

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								AN_PST	TC_PST	DTE_XS_PST	PHY_XS_PST	PCS_PST	WIS_PST	PMD/PMA_PST	CLAUSS22_REG_PST
								r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
15:8	Reserved	Must be kept at reset value.
7	AN_PST	Auto-Negotiation Present In Package Always 1

6	TC_PST	TC Present In Package Always 0
5	DTE_XS_PST	DTE XS Present In Package Always 0
4	PHY_XS_PST	PHY XS Present In Package Always 0
3	PCS_PST	PCS Present In Package Always 0
2	WIS_PST	WIS Present In Package Always 0
1	PMD/PMA_PST	PMD/PMA Present In Package Always 0
0	CLAUSS22_REG_PST	Auto-Negotiation Present In Package Always 1

PCS Package Register 1 (PCS_PKG_1)

Device Address: 0x3
Address offset: 0x06
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDER_SPEC_DEV2_PST	VENDER_SPEC_DEV1_PST	CL22_EXT_PST	Reserved												
r	r	r													

Bits	Fields	Descriptions
15	VENDER_SPEC_DEV2_PST	Vender Specific Device 2 Present in Package Always 0
14	VENDER_SPEC_DEV1_PST	Vender Specific Device 1 Present in Package Always 0
13	CL22_EXT_PST	Clause 22 Extension Present in Package Always 0
12:0	Reserved	Must be kept at reset value.

EEE Capability Register (EEE_CAP)

Device Address: 0x3

Address offset: 0x14

Reset value: 0x0002

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									10GBASE_KR_EEE	10GBASE_KX4_EEE	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	Reserved
									r	r	r	r	r	rw	

Bits	Fields	Descriptions
15:7	Reserved	Must be kept at reset value.
6	10GBASE_KR_EEE	10GBASE-KR EEE Not support, always 0
5	10GBASE_KX4_EEE	10GBASE-KR EEE Not support, always 0
4	1000BASE_KX_EEE	10GBASE-KR EEE Not support, always 0
3	10GBASE_T_EEE	10GBASE-KR EEE Not support, always 0
2	1000BASE_T_EEE	1000Base-T EEE 1: EEE is supported for 1000BASE-T 0: EEE is not supported for 1000Base-T
1	100BASE_TX_EEE	100Base-TX EEE 1: EEE is supported for 100BASE-TX 0: EEE is not supported for 100BASE-TX
0	Reserved	Must be kept at reset value.

EEE Wake Error Counter (EEE_WAKE_ERR_CNT)

Device Address: 0x3

Address offset: 0x16

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEE_WAKE_ERR_CNT[15:0]															
r															

Bits	Fields	Descriptions
15:0	EEE_WAKE_ERR_CNT[15:0]	EEE Wake Error Counter

		Used to count wake time faults where the PHY fails to complete its normal wake sequence within the time required for the specific PHY type. It is cleared when it is read

AN Control Register (AN_CTL)

Device Address: 0x7
Address offset: 0x00
Reset value: 0x1000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AN_RST	Reserved	XNP_CTL	AN_EN	Reserved		RESTART_AUTONEG	Reserved								
rw		r	rw			rw									

Bits	Fields	Descriptions
15	AN_RST	AN Reset Reset the AN and PCS MMD registers to their default value and also cause a software reset in the PHY, it will be selfclearing after reset is finished
14	Reserved	Must be kept at reset value.
13	XNP_CTL	Extended Next Page Control Not support and reserved 0
12	AN_EN	Auto-Negotiation Enable It is a copy of bit 12 in ***PHY control Register (PHY_MII_CTL)*** 1: Auto-Negotiation Enable 0: Auto-Negotiation Disable
11:10	Reserved	Must be kept at reset value.
9	RESTART_AUTONEG	Restart Auto-Negotiation It is a copy of bit 9 in ***PHY control Register (PHY_MII_CTL)***. This bit is self-clearing 1: Restart Auto-Negotiation 0: Normal operation
8:0	Reserved	Must be kept at reset value.

AN Status Register (AN_STS)

Device Address: 0x7
Address offset: 0x01
Reset value: 0x0008

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PAGE_RECEIVED	AN_COMPLETE	AN_ABILITY	AN_ABILITY	LINK_STATUS	Reserved	LP_AN_ABILITY
									r	r	r	r	r		r

Bits	Fields	Descriptions
15:7	Reserved	Must be kept at reset value.
6	PAGE_RECEIVED	Page Received Set to 1 to indicate that a new link codeword has been received and stored in the an_lp_xnp_ability registers. The Page received bit will be reset to 0 on a read of the AN status register or the Auto-Negotiation expansion register. This bit is a copy of bit 1 in ***Auto-Negotiation Expansion Register (PHY_AUTONEG_EXP)***. 1: A new link codeword has been received 0: No link codeword has been received
5	AN_COMPLETE	Auto-Negotiation Complete Set to one to indicate that the Auto-Negotiation has been completed, It will return a 0when Auto-Negotiation is disabled by clearing bit 12 in ***AN Control Register (AN_CTL)***.
4	AN_ABILITY	Remote Fault Set to one to indicate that a remote fault condition has been detected. A remote fault is defined as Far-End-Fault when fiber mode is enable. It is implemented with a latching high function, such that the occurrence of a remote fault causes the bit 4 to become set and remain set until it is cleared. It will be reset to 0 on a read of the AN status register or the Status Register (It is a copy of bit 4 in ***PHY status Register (PHY_MII_STATUS)***).
3	AN_ABILITY	Auto-Negotiation Ability Always set to one in twisted-pair mode to indicate that the PHY has the ability to perform Auto-Negotiation. It is a copy of bit 3 in ***PHY status Register (PHY_MII_STATUS)***
2	LINK_STATUS	Link Status Set to one to indicate that a valid link has been established. It will be implemented with a latching low function, such that the occurrence of a link fail condition causes the link status bit to become cleared and remain cleared until it is read. It is a copy of bit 2 in ***PHY status Register (PHY_MII_STATUS)***
1	Reserved	Must be kept at reset value.
0	LP_AN_ABILITY	Link Partner Auto-Negotiation Ability Set to one to indicate that the link partner is able to participate in the

Auto-Negotiation function

Auto-Negotiation Device Identifier (AN_ID)

Device Address: 0x7

Address offset: 0x02

Reset value: 0x0044

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">PHY_ID [15:0]</td></tr>
<tr><td colspan="16">r</td></tr>
</table>

Bits	Fields	Descriptions
15:0	PHY_ID[15:0]	PHY ID bit[31-16] OUI (bits 3-18). OUI =00-11-05

Auto-Negotiation Device Version Register (AN_VER)

Device Address: 0x7

Address offset: 0x03

Reset value: 0x1400

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="6">PHY_ID[5:0]</td><td colspan="6">PHY_MAN_MODE[5:0]</td><td colspan="4">PHY_REV_NO[3:0]</td></tr>
<tr><td colspan="6">r</td><td colspan="6">r</td><td colspan="4">r</td></tr>
</table>

Bits	Fields	Descriptions
15:10	PHY_ID[5:0]	PHY ID bit [15-10] OUI bits 19-24
9:4	PHY_MAN_MODE[5:0]	Manufacturer’s Model Number Manufacturer’s Model Number (bits 5-0) where [5:4] = architecture version
3:0	PHY_REV_NO[3:0]	Revision Number (bits3-0) *PCS Device Identifier (PCS_ID)* bit 0 is LS bit of PHY Identifier

AN Package Register 0 (AN_PKG_0)

Device Address: 0x7

Address offset: 0x05

Reset value: 0x0089

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	AN_PST	TC_PST	DTE_XS_PST	PHY_XS_PST	PCS_PST	WIS_PST	PMD/PMA_PST	CLAUSS22_REG_PST
	r	r	r	r	r	r	r	r

Bits	Fields	Descriptions
15:8	Reserved	Must be kept at reset value.
7	AN_PST	Auto-Negotiation Present In Package Always 1
6	TC_PST	TC Present In Package Always 0
5	DTE_XS_PST	DTE XS Present In Package Always 0
4	PHY_XS_PST	PHY XS Present In Package Always 0
3	PCS_PST	PCS Present In Package Always 0
2	WIS_PST	PMD/PMA Present In Package Always 0
1	PMD/PMA_PST	PMD/PMA Present In Package Always 0
0	CLAUSS22_REG_PST	Auto-Negotiation Present In Package Always 1

AN Package Register 1 (AN_PKG_1)

Device Address: 0x7
Address offset: 0x06
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VENDER_SPEC_DEV2_PST	VENDER_SPEC_DEV1_PST	CL22_EXT_PST	Reserved												
r	r	r													

Bits	Fields	Descriptions
15	VENDER_SPEC_DEV2_PST	Vender Specific Device 2 Present in Package

		Always 0
14	VENDER_SPEC_DEV1_PST	Vender Specific Device 1 Present in Package Always 0
13	CL22_EXT_PST	Clause 22 Extension Present in Package Always 0
12:0	Reserved	Must be kept at reset value.

Auto-Negotiation Device Identifier (AN_ID)

Device Address: 0x7

Address offset: 0x0E

Reset value: 0x0044

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID [15:0]															
r															

Bits	Fields	Descriptions
15:0	PHY_ID[15:0]	PHY ID bit[31-16] OUI (bits 3-18). OUI =00-11-05

Auto-Negotiation Device Version Register (AN_VER)

Device Address: 0x7

Address offset: 0x0F

Reset value: 0x1400

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_ID[5:0]						PHY_MAN_MODE[5:0]						PHY_REV_NO[3:0]			
r						r						r			

Bits	Fields	Descriptions
15:10	PHY_ID[5:0]	PHY ID bit [15-10] OUI bits 19-24
9:4	PHY_MAN_MODE[5:0]	Manufacturer’s Model Number Manufacturer’s Model Number (bits 5-0) where [5:4] = architecture version
3:0	PHY_REV_NO[3:0]	Revision Number (bits3-0) Register 3, bit 0 is LS bit of PHY Identifier

Auto-Negotiation Advertisement Register (AN_ADV)

Device Address: 0x7

Address offset: 0x10

Reset value: 0x0101

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE_EN	Reserved	REMOTE_FAULT_EN	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE_T4	100BASETX_FULL_DPX	100BASETX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SLCT_FLD[4:0]				
rw		rw	rw	rw	rw	r	rw	rw	rw	rw	r				

Bits	Fields	Descriptions
15	NEXT_PAGE_EN	Next Page Enable 1: Set to use Next Page 0: Not to use Next Page
14	Reserved	Must be kept at reset value
13	REMOTE_FAULT_EN	Remote Fault Detection Enable 1: Auto Negotiation Fault Detected 0: No Remote Fault
12	EXTENED_NEXT_PAGE	Extended Next Page Not supported in the PHY. Should be wrote 0 all the time
11	ASYM_PAUSE	Asymmetric Pause Capability Technology Ability A6 1: Asymmetric Pause capable 0: Asymmetric Pause non-capable
10	PAUSE	Pause Capability Technology Ability A5 1: Pause capable 0: Pause non-capable
9	100BASE_T4	100BASE-T4 Capable Not supported in the PHY. Should be wrote 0 all the time
8	100BASETX_FULL_DPX	100BASE-X Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in *PHY control Register (PHY_MII_CTL)* BIT6, BIT13 (FORCE_SPEED) and BIT8(FORCE_DUPLEX), when FORCE_SPEED is 2'b01 and FORCE_DUPLEX is 1'b1, then this bit will be 1'b1 and vice versa
7	100BASETX_HALF_DPX	100BASE-X Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value

		programmed in *PHY control Register (PHY MII CTL)* BIT6, BIT13 (FORCE_SPEED) and BIT8 (FORCE_DUPLEX). when FORCE_SPEED is 2’b01 and FORCE_DUPLEX is 1’b0, then this bit will be 1’b1 and vice versa
6	10BASE-T_FULL_DPX	10BASE-T Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in *PHY control Register (PHY MII CTL)* BIT6, BIT13 (FORCE_SPEED) and BIT8 (FORCE_DUPLEX). When FORCE_SPEED is 2’b00 and FORCE_DUPLEX is 1’b1, then this bit will be 1’b1 and vice versa
5	10BASE-T_HALF_DPX	10BASE-T Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disabled, value on this bit will reflected the value programmed in *PHY control Register (PHY MII CTL)* BIT6, BIT13 (FORCE_SPEED) and BIT8 (FORCE_DUPLEX). when FORCE_SPEED is 2’b00 and FORCE_DUPLEX is 1’b0, then this bit will be 1’b1 and vice versa
4:0	SLCT_FLD[4:0]	Identifies Type of Message Forced to 5’h01 all the time

Auto-Negotiation Link Partner Ability Register (AN_LP_ABILITY)

Device Address: 0x7

Address offset: 0x13

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEXT_PAGE	ACKNOWLEDGE	REMOTE_FAULT	EXTENED_NEXT_PAGE	ASYM_PAUSE	PAUSE	100BASE T4	100BASE TX_FULL_DPX	100BASE TX_HALF_DPX	10BASE-T_FULL_DPX	10BASE-T_HALF_DPX	SELECTOR_FIELD[4:0]				
r	r	r	r	r	r	r	r	r	r	r	r				

Bits	Fields	Descriptions
15	NEXT_PAGE	Link Partner Next Page Request 1: Link Partner is requesting Next Page function 0: Base Page is requested
14	ACKNOWLEDGE	Link Partner ACKNOWLEDGE Received 1: Link partner acknowledge Received Successfully 0: Not Received
13	REMOTE_FAULT	Link Partner Detects Remote Fault 1: Auto Negotiation Fault Detected 0: No Remote Fault

12	EXTENED_NEXT_PAGE	Extended Next Page
11	ASYM_PAUSE	Link Partner Asymmetric Pause Capable Technology Ability A6 1: Asymmetric Pause capable 0: Asymmetric Pause non-capable
10	PAUSE	Link Partner Symmetric Pause Capable Technology Ability A5 1: Symmetric Pause capable 0: Symmetric Pause non-capable
9	100BASET4	Technology Ability A4 Link Partner 100BASE-T4 Capable
8	100BASETX_FULL_DPX	Link Partner 100BASE-X Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
7	100BASETX_HALF_DPX	Link Partner 100BASE-X Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
6	10BASE-T_FULL_DPX	Link Partner 10BASE-T Full Duplex Capable 1: Capable of Full Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
5	10BASE-T_HALF_DPX	Link Partner 10BASE-T Half Duplex Capable 1: Capable of Half Duplex 0: Not Capable Note: When Auto-Negotiation is disable, value on this bit will be set to 1’b1 all the time
4:0	SELECTOR_FIELD[4:0]	Link Partner Identifies Type of Message Should be 5’h01

Auto-Negotiation XNP Transmit Register (AN_XNP_TRANSMIT)

Device Address: 0x7

Address offset: 0x16

Reset value: 0x2001

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

NEXT_PAGE	Reserved	MESSAGE_PAGE	ACKNOWLEDGE2	TOGGLE	MSG_UFMT_CODE_FIELD[10:0]
rw		rw	rw	r	rw

Bits	Fields	Descriptions
15	NEXT_PAGE	The Last Next Page Indicated whether this is the last next page 1: Additional next page will follow 0: This is the last next page
14	Reserved	Must be kept at reset value.
13	MESSAGE_PAGE	Message Page or Unformatted Page Indicated this is the message page or unformatted page 1: Message Page 0: Unformatted Page
12	ACKNOWLEDGE2	The Ability to Comply with the Message 1: Will comply with the message 0: Can not comply with the message
11	TOGGLE	Toggle The toggle bit will calculated by hardware automatically, SW can ignore
10:0	MSG_UFMT_CODE_FIELD[10:0]	Message/Unformatted Code Field

Auto-Negotiation Link Partner XNP Ability Register (AN_LP_XNP_ABILITY)

Device Address: 0x7

Address offset: 0x19

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NP_RX[15:0]															
r															

Bits	Fields	Descriptions
15	NP_RX[15:0]	Next Page Received from Link Partner

Master-Slave Control Register (MS_CTL)

Device Address: 0x7

Address offset: 0x20

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MS_MAU_CFG_EN	MS_MAU_CFG_VAL	PORT_TYPE	Reserved
rw	rw	rw	

Bits	Fields	Descriptions
15	MS_MAU_CFG_EN	Master-Slave Manual Config Enable 1: Enable MASTER-SLAVE manual configuration 0: Disable MASTER-SLAVE manual configuration
14	MS_MAU_CFG_VAL	Master-Slave Manual Config Value 1: Configure PHY as MASTER 0: Configure PHY as SLAVE
13	PORT_TYPE	Port Type 1: Multi-port device 0: Single-port device
12:0	Reserved	Must be kept at reset value.

Master-Slave Resolution Register (MS_STS)

Device Address: 0x7
Address offset: 0x21
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MS_CFG_FAULT	MS_CFG_RES	Reserved													
r	r														

Bits	Fields	Descriptions
15	MS_CFG_FAULT	Master-Slave Config Fault 1: MASTER-SLAVE configuration fault detected 0: No MASTER-SLAVE configuration fault detected
14	MS_CFG_RES	Master-Slave Config Resolution 1: Local configuration resolved to MASTER 0: Local configuration resolved to SLAVE
13:0	Reserved	Must be kept at reset value.

EEE Advertisement Register (EEE_ADV)

Device Address: 0x7
Address offset: 0x3C

Reset value: 0x0002

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									10GBASE_KR_EEE	10GBASE_KX4_EEE	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	Reserved
									r	r	r	r	rw	rw	

Bits	Fields	Descriptions
15:7	Reserved	Must be kept at reset value.
6	10GBASE_KR_EEE	10GBASE-KR EEE Not support, always 0
5	10GBASE_KX4_EEE	10GBASE-KR EEE Not support, always 0
4	1000BASE_KX_EEE	10GBASE-KR EEE Not support, always 0
3	10GBASE_T_EEE	10GBASE-KR EEE Not support, always 0
2	1000BASE_T_EEE	1000Base-T EEE 1: Advertise that the 1000BASE-T has EEE capability 0: Do not advertise that the 1000BASE-T has EEE capability
1	100BASE_TX_EEE	100Base-TX EEE 1: Advertise that the 100BASE-TX has EEE capability 0: Do not advertise that the 100BASE-TX has EEE capability
0	Reserved	Must be kept at reset value.

EEE Link Partner Ability Register (EEE_LP_ABILITY)

Device Address: 0x7
Address offset: 0x3D
Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									10GBASE_KR_EEE	10GBASE_KX4_EEE	1000BASE_KX_EEE	10GBASE_T_EEE	1000BASE_T_EEE	100BASE_TX_EEE	Reserved
									r	r	r	r	r	r	

Bits	Fields	Descriptions

15:7	Reserved	Must be kept at reset value.
6	10GBASE_KR_EEE	10GBASE-KR EEE 1: Link Partner is advertising EEE capability for 10GBASE-KR 0: Link Partner is not advertising EEE capability for 10GBASEKR EEE
5	10GBASE_KX4_EEE	10GBASE-KX4 EEE 1: Link Partner is advertising EEE capability for 10GBASEKX4 0: Link Partner is not advertising EEE capability for 10GBASEKX4 EEE
4	1000BASE_KX_EEE	1000BASE-KX EEE 1: Link Partner is advertising EEE capability for 1000BASE-KX 0: Link Partner is not advertising EEE capability for 1000BASE-KX EEE
3	10GBASE_T_EEE	10GBASE-T EEE 1: Link Partner is advertising EEE capability for 10GBASE-T 0: Link Partner is not advertising EEE capability for 10GBASET EEE
2	1000BASE_T_EEE	1000BASE-T EEE 1: Link Partner is advertising EEE capability for 1000BASE-T 0: Link Partner is not advertising EEE capability for 1000BASE-T EEE
1	100BASE_TX_EEE	100BASE-TX EEE 1: Link Partner is advertising EEE capability for 100BASE-TX 0: Link Partner is not advertising EEE capability for 100BASE-TX EEE
0	Reserved	Must be kept at reset value.

9. EtherCAT

9.1. Overview

The GDSCN is an EtherCAT SubDevice Controller (ESC). It takes care of the EtherCAT communication as an interface between the EtherCAT fieldbus and the sub application, The GDSCN supports a wide range of applications. The EtherCAT controller has 8K bytes of Process Data RAM(PDRAM) and 8 Fieldbus memory management units (FMMUs), each of which performs the task of mapping logical addresses to physical addresses. The EtherCAT SubDevice controller also includes 8 SyncManagers that allow data exchange between the EtherCAT and the native application. The orientation and mode of operation of each SyncManager is configured by the EtherCAT main device. Two working modes are available: buffer mode and mailbox mode. In buffered mode, the µController and EtherCAT main can write devices simultaneously. The buffer in GDSCN always contains the latest data. If the new data arrives before the old data can be read, the old data will be lost. In mailbox mode, the µController and EtherCAT main access the buffer by shaking hands, ensuring that no data is lost.

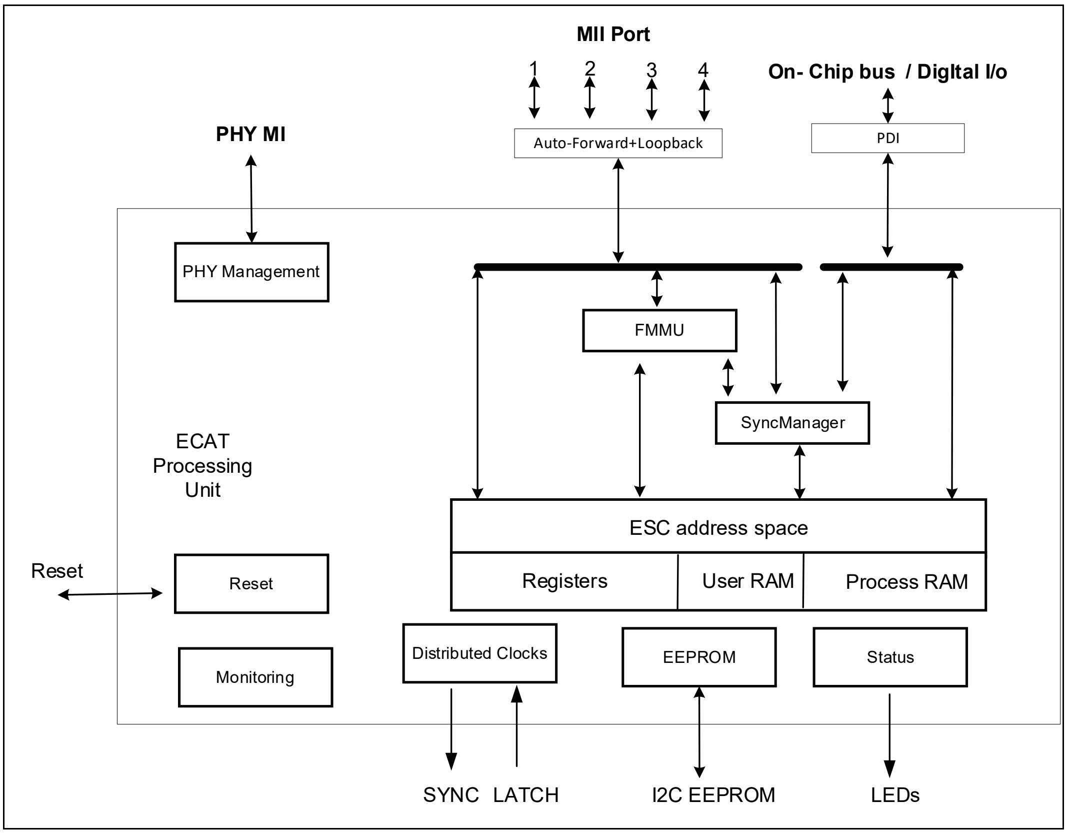
9.2. Characteristics

- Port support: 2 internal phy port and 1 external MII.
- 8 Fieldbus Memory Management Units (FMMUs).
- 8KB PDRAM.
- Distributed clock 64-bit, support allows synchronization with other EtherCAT devices.
- 8 Syncmanager entities.
- DC synchronization less than 1us.

9.2.1. Block diagram

The function module of the ESC is shown in ***Figure 9-1. EtherCAT system block diagram***.

Figure 9-1. EtherCAT system block diagram



9.2.2. EtherCAT SubDevice Controller Function Blocks

- EtherCAT Interfaces

The EtherCAT interfaces or ports connect the ESC to other EtherCAT sub and the main. The MAC layer is integral part of the ESC. The physical layer may be Ethernet. For Ethernet ports, internal Ethernet PHYs connect to the MII ports of the ESC. Transmission speed for EtherCAT is fixed to 100 Mbit/s with Full Duplex communication. Link state and communication status are reported to the Monitoring device. GDSCN uses three ports, port 0/1/2.

- EtherCAT Processing Unit

The EtherCAT Processing Unit (EPU) receives, analyses, and processes the EtherCAT data stream. It is logically located between port 0 and port 3. The main purpose of the EtherCAT Processing unit is to enable and coordinate access to the internal registers and the memory space of the ESC, which can be addressed both from the EtherCAT main and from the local application via the PDI. Data exchange between main and sub application is comparable to a dual-ported memory (process memory), enhanced by special functions e.g. for consistency checking (SyncManager) and data mapping (FMMU). The EtherCAT Processing Units contains the main function blocks of EtherCAT subs besides Auto-Forwarding, Loop-back function, and PDI.

- Auto-Forwarder

The Auto-Forwarder receives the Ethernet frames, performs frame checking and forwards it to the Loop-back function. Time stamps of received frames are generated by the Auto-Forwarder.

- Loop-back function

The Loop-back function forwards Ethernet frames to the next logical port if there is either no link at a port, or if the port is not available, or if the loop is closed for that port. The Loop-back function of port 0 forwards the frames to the EtherCAT Processing Unit. The loop settings can be controlled by the EtherCAT main.

- FMMU

Fieldbus Memory Management Units are used for bitwise mapping of logical addresses to physical addresses of the ESC.

- SyncManager

SyncManagers are responsible for consistent data exchange and mailbox communication between EtherCAT main and subs. The communication direction can be configured for each SyncManager. Read or write transactions may generate events for the EtherCAT main and an attached µController respectively. The SyncManagers are responsible for the main difference between and ESC and a dual-ported memory, because they map addresses to different buffers and block accesses depending on the SyncManager state. This is also a fundamental reason for bandwidth restrictions of the PDI.

- Monitoring

The Monitoring unit contains error counters and watchdogs. The watchdogs are used for observing communication and returning to a safe state in case of an error. Error counters are used for error detection and analysis.

- PHY Management

The PHY Management unit communicates with Ethernet PHYs via the MII management interface. This is either used by the main or by the sub. The MII management interface is used by the ESC itself for optionally restarting auto negotiation after receive errors with the enhanced link detection mechanism, and for the optional MI link detection and configuration feature.

- Distributed Clock

Distributed Clocks (DC) allow for precisely synchronized generation of output signals and input sampling, as well as time stamp generation of events. The synchronization may span the entire EtherCAT network.

- EEPROM

One non-volatile memory is needed for EtherCAT SubDevice Information (ESI) storage, typically an I²C EEPROM.

- Status / LEDs

The Status block provides ESC and application status information. It controls external LEDs like the application RUN LED/ERR LED and port Link/Activity LEDs.

9.3. Function overview

9.3.1. Process Data Interface (PDI)

The Process Data Interface (PDI) realizes the connection between sub application and ESC. Several types of PDIs are defined serial and parallel µController interfaces and Digital I/O interfaces. Due to the high dependency between EtherCAT and PDI accesses to memory, registers, and especially SyncManagers, the internal PDI interface can achieve a maximum throughput of approx 12.5 Mbyte/s.

Table 9-1. PDIs for EtherCAT

PDI number (PDI Control register 0x0140)	PDI name
0	Interface deactivated
4	Digital I/O
128	On-chip bus

PDI Selection and Configuration

Typically, the PDI selection and configuration is part of the ESC Configuration Area of the SII EEPROM. The ESC has the PDI selected and configured at power-on time. In this case, the ESC Configuration Area should reflect the actual settings, although it is not evaluated by the ESC itself. The PDI is active after reset is released, which enables EEPROM emulation by a µController. Take care of Digital Output signals and DC SyncSignals while the EEPROM is not loaded to achieve proper output behavior.

PDI register function acknowledge by write

Some ESC functions are triggered by writing or reading individual byte addresses, SyncManager buffer change or AL event request acknowledge. With an increasing data bus width of the µControllers, this can lead to restrictions or even problems.

Since most µControllers are using byte enable signals for write accesses, there is no restriction for functions which are triggered by writes. But many µControllers are not using the byte enable signals for read accesses, they expect to get a whole data bus width of read data. Reading individual bytes is not possible. This can lead to problems especially by accidentally reading byte addresses which trigger certain ESC functions. Consider a SyncManager buffer area from 0x1000-0x1005. A 32 bit µController application might read the buffer byte-wise. The first access to 0x1000 would open the buffer, and it would also read 0x1001-0x1003. The second access would read 0x1001, and also 0x1000/0x1002-0x1003. The problem occurs when address 0x1004 is to be read, because this would also read 0x1005. The data of 0x1005 is discarded, but the buffer is closed. When the µC reads 0x1005, it will always get 0 – the data seems to be corrupted. A similar issue occurs for DC SyncSignal acknowledging (registers 0x098E and 0x098F). A 32 bit µController would always acknowledge SYNC0 and SYNC1 at the same time, it is not possible to acknowledge them separately.

This problem can be overcome by enabling PDI register function acknowledge by write. In this mode, all functions which are originally triggered by read access are now triggered by corresponding write accesses – which use byte enables and thus can be restricted to certain bytes.

This feature is enabled by IP Core configuration. The current status has to be checked by the µController application in PDI information register 0x014E[0], before using this function.

This feature affects reading of SyncManager buffers and reading of certain registers from PDI side. There is no change to the EtherCAT main side at all. Refer to ***SyncManager*** for SyncManager behavior. The following registers are affected by the PDI register function acknowledge by write feature:

Table 9-2. Registers affected by PDI register function acknowledge by write

Address	name	Trigger function
any	SyncManager buffer end address	Read SyncManager buffer, then write to buffer end address to acknowledge buffer reading.
0x0120:0x0121	AL Control	Read 0x0120:0x0121 after AL Control changes, then write to 0x0120 to acknowledge reading.
0x0440	Watchdog Status Process Data	Read 0x0440, then write to 0x0440 to clear AL event request 0x0220[6]
0x0806+X*16	SyncManager Activate	Read 0x0806+X*16, then write to 0x0806 (SyncManager 0) only to clear AL event request 0x0220[4] for all SyncManagers
0x098E	SYNC0 Status	Read 0x098E, then write to 0x098E to acknowledge DC Sync0 Status 0x098E[0]
0x098F	SYNC1 Status	Read 0x098E, then write to 0x098E to acknowledge DC Sync1 Status 0x098F[0]
0x09B0:0x09B7	Latch0 Time Positive Edge	Read 0x09B0:0x09B7, then write to 0x09B0 to clear DC Latch0 Status 0x09AE[0]
0x09B8:0x09BF	Latch0 Time Negative Edge	Read 0x09B8:0x09BF, then write to 0x09B8 to clear DC Latch0 Status 0x09AE[1]
0x09C0:0x09C7	Latch1 Time Positive Edge	Read 0x09C0:0x09C7, then write to 0x09C0 to clear DC Latch1 Status 0x09AF[0]

0x09C8:0x09CF	Latch1 Time Negative Edge	Read 0x09C8:0x09CF, then write to 0x09C8 to clear DC Latch1 Status 0x09AF[1]

9.3.2. FMMU

Fieldbus Memory Management Units (FMMU) convert logical addresses into physical addresses by the means of internal address mapping. Thus, FMMUs allow to use logical addressing for data segments that span several sub devices: one datagram addresses data within several arbitrarily distributed EtherCAT. Each FMMU channel maps one continuous logical address space to one continuous physical address space of the sub. The access type supported by an FMMU is configurable to be either read, write, or read/write.

■ Restrictions on FMMU Settings

The FMMUs of ESCs are subject to restrictions. The logical address ranges of two FMMUs of the same direction (read or write) in one ESC must be separated by at least 3 logical bytes not configured by any FMMU of the same type, if one of the FMMUs or both use bit-wise mapping (logical start bit ≠ 0, logical stop bit ≠ 7, or physical start bit ≠ 0).

■ Additional FMMU Characteristics

- Each logical address byte can at most be mapped either by one FMMU(read) plus one FMMU(write), or by one FMMU(read/write). If two or more FMMUs (with the same direction – read or write) are configured for the same logical byte, the FMMU with the lower number (lower configuration address space) is used, the other ones are ignored.
- One or more FMMUs may point to the same physical memory, all of them are used. Collisions cannot occur.
- It is the same to use one read/write FMMU or two FMMUs – one read, the other one write – for the same logical address.
- A read/write FMMU cannot be used together with SyncManagers, since independent read and write SyncManagers cannot be configured to use the same (or overlapping) physical address range.
- Bit-wise reading is supported at any address. Bits which are not mapped to logical addresses are not changed in the EtherCAT datagram. E.g., this allows for mapping bits from several ESCs into the same logical byte.
- A frame/datagram addressing a logical address space which is not configured in the ESC will not change data in the ESC, and no data from the ESC is placed in the frame/datagram.

9.3.3. SyncManager

The memory of an ESC can be used for exchanging data between the EtherCAT main and a local application (on a µController attached to the PDI) without any restrictions. Using the memory for communication like this has some drawbacks which are addressed by the SyncManagers inside the ESCs:

- Data consistency is not guaranteed. Semaphores have to be implemented in software for exchanging data in a coordinated way.
- Data security is not guaranteed. Security mechanisms have to be implemented in software.
- Both EtherCAT main and application have to poll the memory in order to find out when the access of the other side has finished.

SyncManagers enable consistent and secure data exchange between the EtherCAT main and the local application, and they generate interrupts to inform both sides of changes.

SyncManagers are configured by the EtherCAT main. The communication direction is configurable, as well as the communication mode (Buffered Mode and Mailbox Mode). SyncManagers use a buffer located in the memory area for exchanging data. Access to this buffer is controlled by the hardware of the SyncManagers.

A buffer has to be accessed beginning with the start address, otherwise the access is denied. After accessing the start address, the whole buffer can be accessed, even the start address again, either as a whole or in several strokes. A buffer access finishes by accessing the end address, the buffer state changes afterwards and an interrupt or a watchdog trigger pulse is generated (if configured). The end address cannot be accessed twice inside a frame.

Two communication modes are supported by SyncManagers:

- Buffered Mode
  - The buffered mode allows both sides, EtherCAT main and local application, to access the communication buffer at any time. The consumer always gets the latest consistent buffer which was written by the producer, and the producer can always update the content of the buffer. If the buffer is written faster than it is read out, old data will be dropped.
  - The buffered mode is typically used for cyclic process data.
- Mailbox Mode
  - The mailbox mode implements a handshake mechanism for data exchange, so that no data will be lost. Each side, EtherCAT main or local application, will get access to the buffer only after the other side has finished its access. At first, the producer writes to the buffer. Then, the buffer is locked for writing until the consumer has read it out. Afterwards, the producer has write access again, while the buffer is locked for the consumer.
  - The mailbox mode is typically used for application layer protocols.

The SyncManagers accept buffer changes caused by the main only if the FCS of the frame is correct, thus, buffer changes take effect shortly after the end of the frame.

The configuration registers for SyncManagers are located beginning at register address 0x0800.

9.3.4. Distributed Clocks

The Distributed Clocks (DC) unit of EtherCAT SubDevice controllers supports the following features:

- Clock synchronization between the subs (and the main)
- Generation of synchronous output signals (SyncSignals)
- Precise time stamping of input events (LatchSignals)
- Generation of synchronous interrupts
- Synchronous Digital Output updates
- Synchronous Digital Input sampling

The device supports 64-bit distributed clocks as detailed in the following sub-sections.

The EtherCAT provides two input pins (SYNC and LATCH) which are used for time stamping of external events. Both rising edge and falling edge time stamps are recorded. These pins are shared with the SYNC0 and LATCH0 output pins, respectively, which are used to indicate the occurrence of time events. The functions of the SYNC/ SYNC0 and LATCH /LATCH0 pins are determined by the SYNC0/LATCH0 Configuration and SYNC/LATCH Configuration bits of the Sync/Latch PDI Configuration Register, respectively.

When set for SYNC0/LATCH0 functionality, the output type (Push-Pull vs. Open Drain/Source) and output polarity are determined by the SYNC0 Output Driver/Polarity and LATCH0 Output Driver/Polarity bits of the Sync/Latch PDI Configuration Register.

9.3.5. EtherCAT State Machine

The EtherCAT State machine (ESM) is responsible for the coordination of main and sub

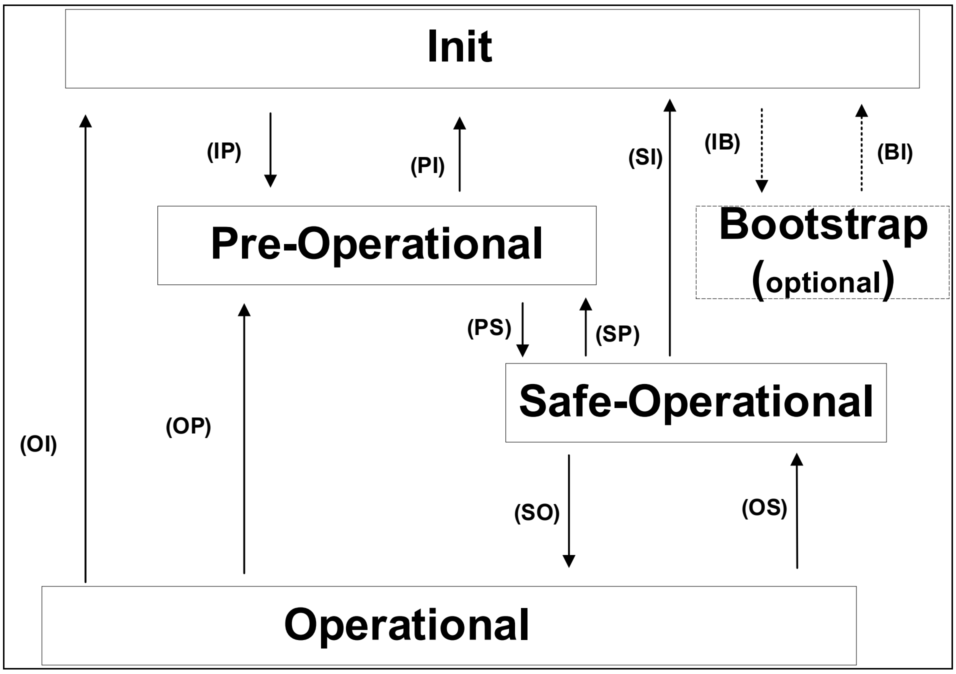
applications at start up and during operation. State changes are typically initiated by requests of the main. They are acknowledged by the local application after the associated operations have been executed. Unsolicited state changes of the local application are also possible.

There are four states an EtherCAT SubDevice shall support, plus one optional state:

- Init (state after Reset)
- Pre-Operational
- Safe-Operational
- Operational
- Bootstrap (optional)

The states and the allowed state changes are shown in ***Figure 9-2. EtherCAT State Machine***:

Figure 9-2. EtherCAT State Machine



NOTE: Not all state changes are possible, the transition from 'Init' to 'Operational' requires the following sequence: Init -> Pre-Operational -> Save-Operational -> Operational.

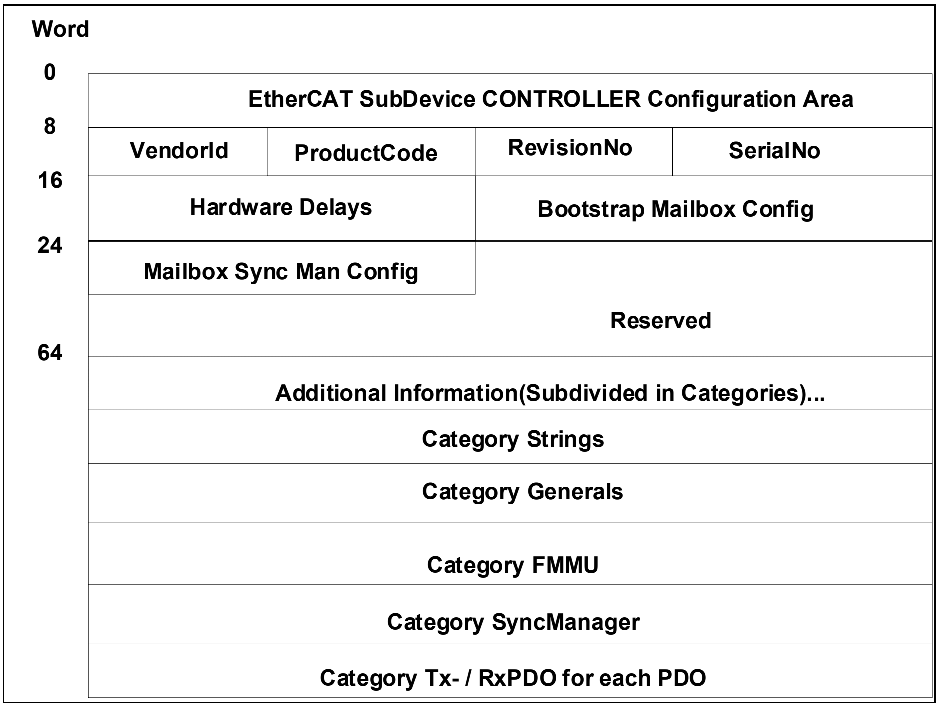
Each state defines required services. Before a state change is confirmed by the sub all services required for the requested state have to be provided or stopped respectively.

9.3.6. EEPROM

EtherCAT SubDevice controllers use a mandatory RAM (typically a serial EEPROM with I²C interface) to store EtherCAT SubDevice Information (ESI). EEPROM sizes from 1 Kbit up to 4 Mbit are supported, depending on the ESC.

The EEPROM structure is shown in ***Figure 9-3. EEPROM Layout*** , the ESI uses word addressing.

Figure 9-3. EEPROM Layout



The following registers are configurable via EEPROM

At least the information stored in the address range from word 0 to 63 (0x00 to 0x3F) is mandatory, as well as the general category (absolute minimum EEPROM size is 2Kbit, complex devices with many categories should be equipped with 32 Kbit EEPROMs or larger). The ESC Configuration area is used by the ESC for configuration. All other parts are used by the main or the local application.

9.3.7. RESET

The EtherCAT module provides two registers, ***ESC Reset register (ESC_RST_ECAT)*** and ***ESC Reset PDI register (ESC_RST_PDI)***, which can be accessed by the EtherCAT main station and sub station respectively
To trigger a reset request.

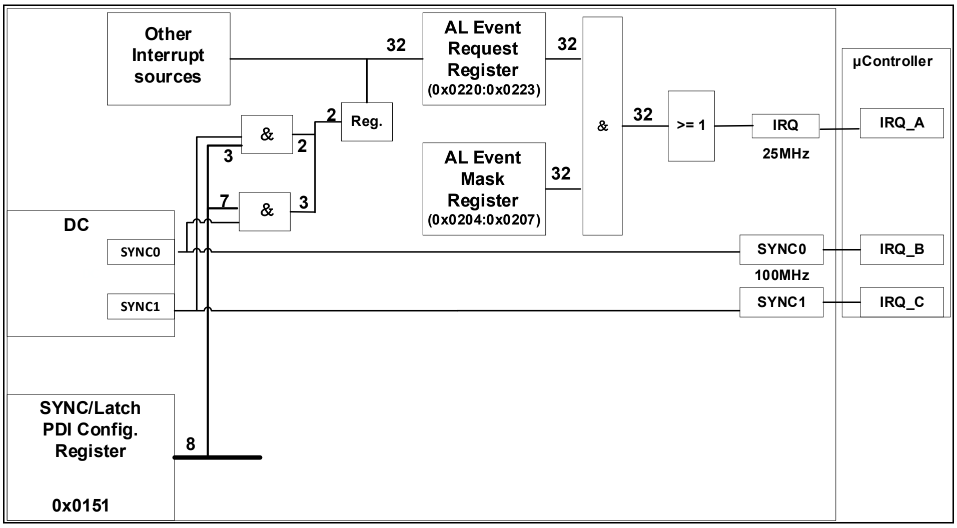
9.3.8. Interrupts

EtherCAT support two types of interrupts: AL Event Requests targeted at a µController, and EtherCAT event requests targeted at the EtherCAT main. Additionally, the Distributed Clocks SyncSignals can be used as interrupts for a µController as well.

AL Event Request (PDI Interrupt)

AL Event Requests can be signaled to a µController using the PDI Interrupt Request signal (IRQ/SPI_IRQ, etc.). for IRQ generation, the AL Event Request register (0x0220:0x0223) is combined with the AL Event Mask register (0x0204:0x0207) using a logical AND operation, then all resulting bits are combined (logical OR) into one interrupt signal. The output driver characteristics of the IRQ signal are configurable using the SYNC/LATCH PDI configuration register (0x0151). The AL Event Mask register allows for selecting the interrupts which are relevant for the µController and handled by the application.

Figure 9-4. PDI Interrupt Masking and interrupt signals



The DC SyncSignals can be used for interrupt generation in two ways:

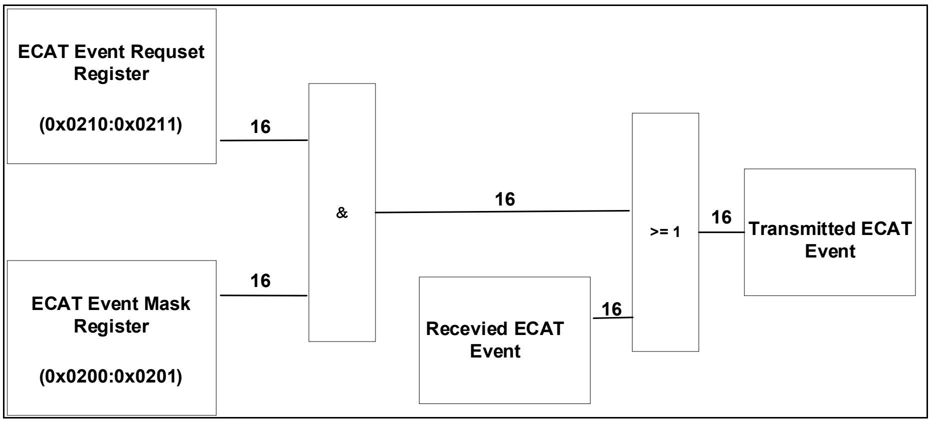
- The DC SYNC signals are mapped into the AL Event Request Register (configured with SYNC/LATCH PDI Configuration register 0x0151.3/7). In this case, all interrupts from the ESC to the µController are combined into one IRQ signal, and the Distributed Clocks LATCH0/1 inputs can still be used. The IRQ signal has a jitter of ~40 ns.
- The DC SyncSignals are directly connected to µController interrupt inputs. The µController can react on DC SyncSignal interrupts faster (without reading AL Request register), but it needs more interrupt inputs. The jitter of the SyncSignals is ~12 ns. The DC Latch functions are only available for one Latch input or not at all (if both DC SYNC outputs are used).

ECAT Event Request (ECAT Interrupt)

ECAT event requests are used to inform the EtherCAT main of sub events. ECAT events make use of the IRQ field inside EtherCAT datagrams. The ECAT Event Request register (0x0210:0x0211) is combined with the ECAT Event Mask register (0x0200:0x0201) using a logical AND operation. The resulting interrupt bits are combined with the incoming ECAT IRQ field using a logical OR operation, and written into the outgoing ECAT IRQ field. The ECAT Event Mask register allows for selecting the interrupts which are relevant for the EtherCAT main and handled by the main application.

NOTE: The main cannot distinguish which sub (or even more than one) was the origin of an interrupt.

Figure 9-5. EtherCAT Interrupt Masking



Clearing Interrupts Accidentally

Event request registers and register actions which clear interrupts are intended to be accessed independently, i.e., with separate EtherCAT frames or separate PDI accesses. Otherwise it may happen that interrupts and/or data are missed.

9.3.9. LED

EtherCAT SubDevice controllers support LED(RUNLED) regarding link state and AL status. The LED output of an ESC is controlled by the AL status register (0x0130) and supports the following states, which are automatically translated into blink codes.

The EtherCAT Core configuration provides for direct control of the RUN LED via the RUN LED Override Register.

9.4. ESC Register definition

9.4.1. ESC Type register (ESC_TYPE)

Address Offset: 0x0000
Reset value: 0xBC

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
TYPE[7:0]							
r(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	TYPE[7:0]	Type of EtherCAT controller 0xBC: GDSCN

9.4.2. ESC Revision register (ESC_REVISION)

Address Offset: 0x0001

Reset value: 0x0000

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
REVISION[7:0]							
r(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	REVISION[7:0]	Revision of EtherCAT controller.

9.4.3. ESC Build register (ESC_BUILD)

Address Offset: 0x0002

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PATCHBUILD[7:0]								MINVY[3:0]				MAINVZ[3:0]			
r(ECAT)								r(ECAT)				r(ECAT)			
r(PDI)								r(PDI)				r(PDI)			

Bits	Fields	Descriptions
15:8	PATCHBUILD[7:0]	patch level / development build: 0x00: original release 0x01-0x0F: patch level of original release 0x10-0xFF: development build
7:4	MINVY[3:0]	minor version Y
3:0	MAINVZ[3:0]	maintenance version Z

9.4.4. ESC FMMU Numbers register (ESC_FMMUS)

Address Offset: 0x0004

Reset value: 0x08

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
FMMU[7:0]							
r(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	FMMU[7:0]	This field details the number of supported FMMU channels (or entities) of the EtherCAT SubDevice controller. The device provides 8.

9.4.5. ESC SyncManagers Numbers register (ESC_SYNCMS)

Address Offset: 0x0005

Reset value: 0x08

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
SM [7:0]							
r(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	SM[7:0]	Number of supported SyncManager entities, The device provides 8.

9.4.6. ESC RAM size register (ESC_RAMSIZE)

Address Offset: 0x0006
Reset value: 0x08

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
RAMSIZE[7:0]							
r(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	RAMSIZE[7:0]	Process Data RAM size supported in 8Kbyte

9.4.7. ESC Port Descriptor register (ESC_PORT_DESC)

Address Offset: 0x0007
Reset value: 0x3F

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Port3[1:0]		Port2[1:0]		Port1[1:0]		Port0[1:0]	
r(ECAT)		r(ECAT)		r(ECAT)		r(ECAT)	
r(PDI)		r(PDI)		r(PDI)		r(PDI)	

Bits	Fields	Descriptions
7:6	Port3[1:0]	Port 3 Configuration This field details the Port 3 configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII
5:4	Port2[1:0]	Port 2 Configuration This field details the Port 2configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII
3:2	Port1[1:0]	Port 1 Configuration This field details the Port 1 configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII
1:0	Port0[1:0]	Port 0 Configuration This field details the Port 0 configuration. 00: Not implemented 01: Not configured 10: Reserved 11: MII

9.4.8. ESC Features supporter register (ESC_FEATURE)

Address Offset: 0x0008

Reset value: 0x01CC

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				FS11	FS10	FS9	FS8	FS7	FS6	Reserved		FS3	FS2	FS1	FS0
				r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)			r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
				r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)			r(PDI)	r(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
15:12	Reserved	Must be kept at reset value.
11	FS11	Fixed FMMU/SyncManager configuration: 0: Variable configuration

		1: Fixed configuration
10	FS10	EtherCAT read/write command support (BRW, APRW, FPRW): 0: Supported 1: Not supported
9	FS9	EtherCAT LRW command support: 0: Supported 1: Not supported
8	FS8	Enhanced DC SYNC Activation: 0: Not available 1: Available NOTE: This feature refers to registers *ESC Register Activation register (ESC_REG_ACT)* and *ESC Activation Status register (ESC_ACT_STAT)*
7	FS7	Separate Handling of FCS Errors: 0: Not supported 1: Supported, frames with wrong FCS and additional nibble will be counted separately in Forwarded RX Error Counter
6	FS6	Enhanced Link Detection MII: 0: Not available 1: Available
5:4	Reserved	Must be kept at reset value.
3	FS3	Distributed Clocks (width): 0: 32 bit 1: 64 bit
2	FS2	Distributed Clocks: 0: Not available 1: Available
1	FS1	Unused register access: 0: allowed 1: not supported
0	FS0	FMMU Operation 0: Bit oriented 1: Byte oriented

9.4.9. ESC Configured station address register (ESC_STATION_ADDR)

Address Offset: 0x0010
Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSAD[15:0]															
rw(ECAT) r(PDI)															

Bits	Fields	Descriptions
15:0	CSAD[15:0]	Address used for node addressing (FPRD/FPWR/FPRW/FRMW commands).

9.4.10. ESC Configured station Alias register (ESC_STATION_ALS)

Address Offset: 0x0012

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSAL[15:0]															
r(ECAT) rw(PDI)															

Bits	Fields	Descriptions
15:0	CSAL[15:0]	Alias Address used for node addressing (FPRD/FPWR/FPRW/FRMW commands). The use of this alias is activated by Register ***ESC DL Control register (ESC_DL_CTL)*** Bit [24]. NOTE: EEPROM value is only transferred into this register at first EEPROM load after power-on or reset.

9.4.11. Write Enable register (ESC_REG_WEN)

Address Offset: 0x0020

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							WEN

							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	WEN	If register write protection is enabled, this register has to be written in the same Ethernet frame (value does not matter) before other writes to this station are allowed. This bit is self-clearing at the beginning of the next frame (SOF), or if

Register Write Protection is disabled.

9.4.12. ESC Write Protection register (ESC_REG_WP)

Address Offset: 0x0021

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Resrved							WPR
							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	WPR	Register write protection: 0: Protection disabled 1: Protection enabled Registers 0x0000:0x0F7F are write-protected, except for 0x0020 and 0x0030.

9.4.13. ESC Write Enable register (ESC_WEN)

Address Offset: 0x0030

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Resrved							EWEN
							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	EWEN	If ESC write protection is enabled, this register has to be written in the same Ethernet frame (value does not matter) before other writes to this station are allowed. This bit is self-clearing at the beginning of the next frame (SOF), or if ESC Write Protection is disabled.

9.4.14. ESC Write Protection register (ESC_WP)

Address Offset: 0x0031

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Resrved							EWPR

							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	EWPR	Write protect: 0: Protection disabled 1: Protection enabled All areas are write-protected, except for 0x0030.

9.4.15. ESC Reset register (ESC_RST_ECAT)

Address Offset: 0x0040
Reset value: 0x0

This register can be accessed by byte(8-bit).

Write:

7	6	5	4	3	2	1	0
RESET_ECAT[7:0]							

rw(ECAT)
r(PDI)							

Bits	Fields	Descriptions
7:0	RESET_ECAT[7:0]	A reset is asserted after writing the reset sequence 0x52 (‘R’), 0x45 (‘E’) and 0x53 (‘S’) in this register with 3 consecutive frames. Any other frame which does not continue the sequence by writing the next expected value will cancel the reset procedure.

Read:

7	6	5	4	3	2	1	0
Reserved						RESET_ECAT[1:0]	

						rw(ECAT)
						r(PDI)	

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1:0	RESET_ECAT[1:0]	Progress of the reset procedure: 00: initial/reset state 01: after writing 0x52 (‘R’), when previous state was 00 10: after writing 0x45 (‘E’), when previous state was 01

11: after writing 0x53 (‘S’), when previous state was 10.

This value must not be observed because the ESC enters reset when this state is reached, resulting in state 00.

9.4.16. ESC Reset PDI register (ESC_RST_PDI)

Address Offset: 0x0041

Reset value: 0x0

This register can be accessed by byte(8-bit).

Write:

7	6	5	4	3	2	1	0
RESET_PDI[7:0]							
r(ECAT)							
rw(PDI)							

Bits	Fields	Descriptions
7:0	RESET_PDI[7:0]	A reset is asserted after writing the reset sequence 0x52 (‘R’), 0x45 (‘E’) and 0x53 (‘S’) in this register with 3 consecutive commands. Any other command which does not continue the sequence by writing the next expected value will cancel the reset procedure.

Read:

7	6	5	4	3	2	1	0
Reserved						RESET_PDI[1:0]	
						r(ECAT)	
						rw(PDI)	

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1:0	RESET_PDI[1:0]	Progress of the reset procedure: 00: initial/reset state 01: after writing 0x52 (‘R’), when previous state was 00 10: after writing 0x45 (‘E’), when previous state was 01 11: after writing 0x53 (‘S’), when previous state was 10. This value must not be observed because the ESC enters reset when this state is reached, resulting in state 00.

9.4.17. ESC DL Control register (ESC_DL_CTL)

Address Offset: 0x0100

Reset value: 0x7C001

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							ALIAS	Reserved					FIFO_REDUCTION[2:0]		
							rw(ECAT)							rw(ECAT)	
							r(PDI)							r(PDI)	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOOP_P3[1:0]		LOOP_P2[1:0]		LOOP_P1[1:0]		LOOP_P0[1:0]		Reserved						USE	FRAME
rw(ECAT)		rw(ECAT)		rw(ECAT)		rw(ECAT)								rw(ECAT)	rw(ECAT)
r(PDI)		r(PDI)		r(PDI)		r(PDI)								r(PDI)	r(PDI)

Bits	Fields	Descriptions
31:25	Reserved	Must be kept at reset value.
24	ALIAS	Station alias: 0: Ignore Station Alias 1: Alias can be used for all configured address command types (FPRD, FPWR, …)
23:19	Reserved	Must be kept at reset value.
18:16	FIFO_REDUCTION[2:0]	RX FIFO Size (ESC delays start of forwarding until FIFO is at least half full). RX FIFO Size/RX delay reduction: Value: MII: 0: -40 ns (-80 ns) 1: -40 ns (-80 ns) 2: -40 ns 3: -40 ns 4: no change 5: no change 6: no change 7: default NOTE: EEPROM value is only taken over at first EEPROM load after power-on or reset
15:14	LOOP_P3[1:0]	Loop Port 3: 00: Auto 01: Auto Close 10: Open 11: Closed
12:13	LOOP_P2[1:0]	Loop Port 2: 00: Auto 01: Auto Close 10: Open 11: Closed
11:10	LOOP_P1[1:0]	Loop Port 1:

		00: Auto 01: Auto Close 10: Open 11: Closed
9:8	LOOP_P0[1:0]	Loop Port 0: 00: Auto 01: Auto Close 10: Open 11: Closed NOTE: Loop open means sending/receiving over this port is enabled, loop closed means sending/receiving is disabled and frames are forwarded to the next open port internally. Auto: loop closed at link down, opened at link up Auto Close: loop closed at link down, opened with writing 01 again after link up (or receiving a valid Ethernet frame at the closed port) Open: loop open regardless of link state Closed: loop closed regardless of link state
7:2	Reserved	Must be kept at reset value.
1	USE	Temporary use of settings in 0x0100:0x0103[8:15]: 0: permanent use 1: use for about 1 second, then revert to previous settings
0	FRAME	Forwarding rule: 0: Forward non-EtherCAT frames: EtherCAT frames are processed, non-EtherCAT frames are forwarded without processing or modification. The source MAC address is not changed for any frame. 1: Destroy non-EtherCAT frames: EtherCAT frames are processed, non-EtherCAT frames are destroyed. The source MAC address is changed by the Processing Unit for every frame (SOURCE_MAC[1] is set to 1 – locally administered address).

9.4.18. ESC Physical read/write offset register (ESC_PHYSICAL_RW_OFFSET)

Address Offset: 0x0108

Reset value: 0x0

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">RW_OFFSET[15:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions

15:0	RW_OFFSET[15:0]	This register is used for ReadWrite commands in Device Addressing mode (FPRW, APRW, BRW). The internal read address is directly taken from the offset address field of the EtherCAT datagram header, while the internal write address is calculated by adding the Physical Read/Write Offset value to the offset address field. Internal read address = ADR, internal write address = ADR + R/W-Offset

9.4.19. ESC DL Status register (ESC_DL_STAT)

Address Offset: 0x0110

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL_STAT US15	DL_STAT US14	DL_STAT US13	DL_STAT US12	DL_STAT US1`	DL_STAT US10	DL_STAT US9	DL_STAT US8	DL_STAT US7	DL_STAT US6	DL_STAT US5	DL_STAT US4	Reserved	DL_STAT US2	DL_STAT US1	DL_STAT US0
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	Fields	Descriptions
15	DL_STATUS15	Communication on Port 3: 0: No stable communication 1: Communication established
14	DL_STATUS14	Loop Port 3: 0: Open 1: Closed
13	DL_STATUS13	Communication on Port 2: 0: No stable communication 1: Communication established
12	DL_STATUS12	Loop Port 2: 0: Open 1: Closed
11	DL_STATUS11	Communication on Port 1: 0: No stable communication 1: Communication established
10	DL_STATUS10	Loop Port 1: 0: Open 1: Closed
9	DL_STATUS9	Communication on Port 0: 0: No stable communication 1: Communication established
8	DL_STATUS8	Loop Port 0:

		0: Open 1: Closed
7	DL_STATUS7	Physical link on Port 3 0: No link 1: Link detected
6	DL_STATUS6	Physical link on Port 2 0: No link 1: Link detected
5	DL_STATUS5	Physical link on Port 1 0: No link 1: Link detected
4	DL_STATUS4	Physical link on Port 0: 0: No link 1: Link detected
3	Reserved	Must be kept at reset value.
2	DL_STATUS2	Enhanced Link detection: 0: Deactivated for all ports 1: Activated for at least one port NOTE: EEPROM value is only transferred into this register at first EEPROM load after power-on or reset
1	DL_STATUS1	PDI Watchdog Status: 0: Watchdog expired 1: Watchdog reloaded
0	DL_STATUS0	PDI operational/EEPROM loaded correctly: 0: EEPROM not loaded, PDI not operational (no access to Process Data RAM) 1: EEPROM loaded correctly, PDI operational (access to Process Data RAM)

9.4.20. ESC AL Control register (ESC_AL_CTL)

Address Offset: 0x0120

Reset value: 0x1

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										REQUEST	ACK	STATE[3:0]			
										rw(ECAT)	rw(ECAT)	rw(ECAT)			
										rw(PDI)	rw(PDI)	rw(PDI)			

Bits	Fields	Descriptions
15:6	Reserved	Must be kept at reset value.
5	REQUEST	Device Identification:

		0: No request 1: Device Identification request
4	ACK	Error Ind Ack: 0: No Ack of Error Ind in AL status register 1: Ack of Error Ind in AL status register
3:0	STATE[3:0]	Initiate State Transition of the Device State Machine: 1: Request Init State 3: Request Bootstrap State 2: Request Pre-Operational State 4: Request Safe-Operational State 8: Request Operational State Note: PDI register function acknowledge by Write command is disabled: Reading AL Control from PDI clears AL Event Request 0x0220[0]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing AL Control from PDI clears AL Event Request 0x0220[0]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: AL Control register behaves like a mailbox if Device Emulation is off (0x0141[0]=0): The PDI has to read/write* the AL Control register after ECAT has written it. Otherwise ECAT cannot write again to the AL Control register. After Reset, AL Control register can be written by ECAT. (Regarding mailbox functionality, both low and high byte of the AL Control register trigger read/write functions, e.g., reading 0x0121 is sufficient to make this register writable again) If Device Emulation is on, the AL Control register can always be written, its content is copied to the AL Status register.

9.4.21. ESC AL Status register (ESC_AL_STAT)

Address Offset: 0x0130

Reset value: 0x1

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										IDENTI	ERROR_IND	AL_STATE[3:0]			
										r(ECAT) rw(PDI)	r(ECAT) rw(PDI)	r(ECAT) rw(PDI)			

Bits	Fields	Descriptions
15:6	Reserved	Must be kept at reset value.
5	IDENTI	Device Identification: 0: Device Identification not valid 1: Device Identification loaded
4	ERROR_IND	Error Ind:

		0: Device is in State as requested or Flag cleared by command 1: Device has not entered requested State or changed State as result of a local action
3:0	AL_STATE[3:0]	Actual State of the Device State Machine: 1: Init State 3: Bootstrap State 2: Pre-Operational State 4: Safe-Operational State 8: Operational State

NOTE: AL Status register is only writable from PDI if Device Emulation is off (0x0141[0]=0), otherwise AL Status register will reflect AL Control register values. Avoid reading AL Status register from PDI.

9.4.22. ESC AL Status Code register (ESC_AL_STAT_CODE)

Address Offset: 0x0134
Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL_STATUS_CODE[15:0]															
r(ECAT) rw(PDI)															

Bits	Fields	Descriptions
15:0	AL_STATUS_CODE[15:0]	AL Status Code

9.4.23. ESC RUN LED Override register (ESC_RUN_LED_OVRD)

Address Offset: 0x0138
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved			OVERRIDEN	LED_CODE[3:0]			

			rw(ECAT)
			rw(PDI)
				rw(ECAT)
				rw(PDI)			

Bits	Fields	Descriptions
7:5	Reserved	Must be kept at reset value.
4	OVERRIDEN	Enable Override:

		0: Override disabled 1: Override enabled	
3:0	LED_CODE[3:0]	LED code:	AL Status:
		0x0: Off	Init 1
		0x1: Flash 1x	
		0x2-0xC: Flash 2x – 12x	SafeOp 4
		0xD: Blinking	PreOp 2
		0xE: Flickering	Bootstrap 3
		0xF: On	Operational 8

NOTE: Changes to AL Status register (0x0130) with valid values will disable RUN LED Override (0x0138[4]=0). The value read in this register always reflects current LED output.

9.4.24. ESC ERR LED Override register (ESC_ERR_LED_OVRD)

Address Offset: 0x0139
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved			OVERRIDEN	LED_CODE[3:0]			

			rw(ECAT)
			rw(PDI)
				rw(ECAT)
				rw(PDI)			

Bits	Fields	Descriptions
7:5	Reserved	Must be kept at reset value.
4	OVERRIDEN	Enable Override: 0: Override disabled 1: Override enabled
3:0	LED_CODE[3:0]	LED code: 0x0: Off 0x1-0xC: Flash 1x – 12x 0xD: Blinking 0xE: Flickering 0xF: On

NOTE: New error conditions will disable ERR LED Override (0x0139[4]=0). The value read in this register always reflects current LED output.

9.4.25. ESC PDI Control register (ESC_PDI_CTL)

Address Offset: 0x0140

Reset value: depends on configuration

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
PDI_CTL[7:0]							
r(ECAT)							
r(PDI)							

Bits	Fields	Descriptions
7:0	PDI_CTL[7:0]	Process data interface: 0x04: Digital I/O 0x80: PDI Select SPI or EXMC, When the pad of MCU_PDI_TYPE is 1, select EXMC; When the pad of MCU_PDI_TYPE is 0, select SPI. Others: Reserved

9.4.26. ESC Configuration register (ESC_CFG)

Address Offset: 0x0141

Reset value: depends on configuration

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
EH_LINK_P3	EH_LINK_P2	EH_LINK_P1	EH_LINK_P0	CLK_LATCH_EN	CLK_SYNC_EN	EH_LINK_ALL	CTL_AL_STATUS
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
7	EH_LINK_P3	Enhanced Link port 3(default 1, later EEPROM word 0): 0: disabled (if bit 1=0) 1: enabled
6	EH_LINK_P2	Enhanced Link port 2(default 1, later EEPROM word 0): 0: disabled (if bit 1=0) 1: enabled
5	EH_LINK_P1	Enhanced Link port 1(default 1, later EEPROM word 0): 0: disabled (if bit 1=0) 1: enabled
4	EH_LINK_P0	Enhanced Link port 0(default 1, later EEPROM word 0): 0: disabled (if bit 1=0)

		1: enabled
3	CLK_LATCH_EN	Distributed Clocks Latch In Unit: 0: disabled (power saving) 1: enabled
2	CLK_SYNC_EN	Distributed Clocks SYNC Out Unit: 0: disabled (power saving) 1: enabled
1	EH_LINK_ALL	Enhanced Link detection all ports(default 1, later EEPROM word 0): 0: disabled (if bits [7:4]=0) 1: enabled at all ports (overrides bits [7:4])
0	CTL_AL_STATUS	Device emulation (control of AL status) (default 0) 0: AL status register has to be set by PDI 1: AL status register will be set to value written to AL control register NOTE: Reset value is 1 with Digital I/O PDI，On-chip bus Others: 0。

NOTE: EEPROM values of bits 1, 4, 5, 6, and 7 are only transferred into this register at first EEPROM load after power-on or reset.

9.4.27. ESC PDI Information register (ESC_PDI_INFO)

Address Offset: 0x014E

Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												CONFIGURE_EN	ACTIVE_EN	LOAD_EN	PDI_WREN
												r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
												r(PDI)	r(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
15:4	Reserved	Must be kept at reset value.
3	CONFIGURE_EN	PDI configuration invalid(default 0): 0: PDI configuration ok 1: PDI configuration invalid
2	ACTIVE_EN	PDI active(default 0): 0: PDI not active 1: PDI active
1	LOAD_EN	ESC configuration area loaded from EEPROM(default 0): 0: not loaded 1: loaded

0	PDI_WREN	PDI function acknowledge by write: 0: Disabled 1: Enabled Note: Reset value depends on configuration

9.4.28. ESC PDI configuration register (ESC_PDI_CFG)

Address Offset: 0x0150

Reset value: depends on configuration

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
OUT_DATA[1:0]		IN_DATA[1:0]		WATCHDOG	MODE	OUTVALID_MOD	POLARITY
r(ECAT)		r(ECAT)		r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
r(PDI)		r(PDI)		r(PDI)	r(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
7:6	OUT_DATA[1:0]	Output DATA is updated at 00: End of Frame 01: reserved 10: DC SYNC0 event 11: DC SYNC1 event If 0x0150[1]=1, output DATA is updated at Process Data Watchdog trigger event (0x0150[7:6] are ignored)
5:4	IN_DATA[1:0]	Input DATA is sampled at 00: Start of Frame2 01: Rising edge of LATCH_IN 10: DC SYNC0 event 11: DC SYNC1 event
3	WATCHDOG	Watchdog behavior(default 0): 0: Outputs are reset immediately after watchdog expires 1: Outputs are reset with next output event that follows watchdog expiration
2	MODE	Unidirectional/Bidirectional mode(default 0): 0: Unidirectional mode: input/output direction of pins configured individually 1: Bidirectional mode: all I/O pins are bidirectional, direction configuration is ignored
1	OUTVALID_MOD	OUTVALID mode(default 0): 0: Output event signaling 1: Process Data Watchdog trigger (WD_TRIG) signaling on OUTVALID pin (see SyncManager). Output data is updated if watchdog is triggered. Overrides 0x0150[7:6]
0	POLARITY	OUTVALID polarity(default 0):

0: Active high

1: Active low

NOTE: all the bit can be configured via EEPROM.

9.4.29. ESC Sync/Latch configuration register (ESC_SL_CFG)

Address Offset: 0x0151

Reset value: depends on configuration

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
SYNC1_MAP	SL1_CONFIG	SYNC1_POLARITY[0:1]		SYNC0_MAP	SL0_CONFIG	SYNC1_POLARITY[1:0]	
r(ECAT)	r(ECAT)	r(ECAT)		r(ECAT)	r(ECAT)	r(ECAT)	
r(PDI)	r(PDI)	r(PDI)		r(PDI)	r(PDI)	r(PDI)	

Bits	Fields	Descriptions
7	SYNC1_MAP	SYNC1 mapped to AL Event Request Register 0x0220[3]: 0: Disabled 1: Enabled
6	SL1_CONFIG	SYNC1/LATCH1 Configuration(default 1): 0: LATCH1 Input 1: SYNC1 Output NOTE: The ESC has concurrent SYNC[1:0] outputs and LATCH[1:0] inputs, independent of this configuration
5:4	SYNC1_POLARITY[0:1]	SYNC1 Output Driver/Polarity(default 10): 00: Push-Pull (Active Low) 01: Open Drain (Active Low) 10: Push-Pull (Active High) 11: Open Source (Active High)
3	SYNC0_MAP	SYNC0 mapped to AL Event Request Register 0x0220[2]: 0: Disabled 1: Enabled
2	SL0_CONFIG	SYNC0/LATCH0 Configuration(default 1): 0: LATCH0 Input 1: SYNC0 Output NOTE: The ESC has concurrent SYNC[1:0] outputs and LATCH[1:0] inputs, independent of this configuration
1:0	SYNC1_POLARITY[1:0]	SYNC0 Output Driver/Polarity(default 10): 00: Push-Pull (Active Low) 01: Open Drain (Active Low)

10: Push-Pull (Active High)

11: Open Source (Active High)

9.4.30. ESC PDI extended configuration register (ESC_PDI_EXT_CFG)

Address Offset: 0x0152

Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR15	DIR15	DIR15	DIR15	DIR15	DIR15	DIR15	DIR8	DIR7	DIR6	DIR5	DIR4	DIR3	DIR2	DIR1	DIR0
r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)	r(ECAT)
r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
15	DIR15	Direction of I/O[31:30] configured in pairs as inputs or outputs: Refer to DIR0 description
14	DIR14	Direction of I/O[29:28] configured in pairs as inputs or outputs: Refer to DIR0 description
13	DIR13	Direction of I/O[27:26] configured in pairs as inputs or outputs: Refer to DIR0 description
12	DIR12	Direction of I/O[25:24] configured in pairs as inputs or outputs: Refer to DIR0 description
11	DIR11	Direction of I/O[23:22] configured in pairs as inputs or outputs: Refer to DIR0 description
10	DIR10	Direction of I/O[21:20] configured in pairs as inputs or outputs: Refer to DIR0 description
9	DIR9	Direction of I/O[19:18] configured in pairs as inputs or outputs: Refer to DIR0 description
8	DIR8	Direction of I/O[17:16] configured in pairs as inputs or outputs: Refer to DIR0 description
7	DIR7	Direction of I/O[15:14] configured in pairs as inputs or outputs: Refer to DIR0 description
6	DIR6	Direction of I/O[13:12] configured in pairs as inputs or outputs: Refer to DIR0 description
5	DIR5	Direction of I/O[11:10] configured in pairs as inputs or outputs: Refer to DIR0 description
4	DIR4	Direction of I/O[9:8] configured in pairs as inputs or outputs:

		Refer to DIR0 description
3	DIR3	Direction of I/O[7:6] configured in pairs as inputs or outputs: Refer to DIR0 description
2	DIR2	Direction of I/O[5:4] configured in pairs as inputs or outputs: Refer to DIR0 description
1	DIR1	Direction of I/O[3:2] configured in pairs as inputs or outputs: Refer to DIR0 description
0	DIR0	Direction of I/O[1:0] configured in pairs as inputs or outputs: 0: Input 1: Output NOTE: Reserved in bidirectional mode, set to 0. Configuration bits for unavailable I/Os are reserved, set EEPROM value to 0.

9.4.31. ESC Event Mask register (ESC_EVT_MSK)

Address Offset: 0x0200

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT_REQ[15:0]															

rw(ECAT)

r(PDI)

Bits	Fields	Descriptions
15:0	EVENT_REQ[15:0]	ECAT Event masking of the ECAT Event Request Events for mapping into ECAT event field of EtherCAT frames: 0: Corresponding ECAT Event Request register bit is not mapped 1: Corresponding ECAT Event Request register bit is mapped

9.4.32. ESC PDI AL Event register (ESC_PDI_AL_EVT)

Address Offset: 0x0204

Reset value: 0x00FFFF0F

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AL_EVENT_REQ[31:16]															

r(ECAT)

rw(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AL_EVENT_REQ[15:0]															

r(ECAT)

rw(PDI)

Bits	Fields	Descriptions
31:0	AL_EVENT_REQ[31:0]	AL Event masking of the AL Event Request register Events for mapping to PDI IRQ signal: 0: Corresponding AL Event Request register bit is not mapped 1: Corresponding AL Event Request register bit is mapped

9.4.33. ESC Event Request register (ESC_EVT_REQ)

Address Offset: 0x0210

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CHANNEL 7	CHANNEL 6	CHANNEL 5	CHANNEL 4	CHANNEL 3	CHANNEL 2	CHANNEL 1	CHANNEL 0	AL_STAT US	DL_STAT US	Reserved	DC_LATC H
				r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)

Bits	Fields	Descriptions
15:12	Reserved	Must be kept at reset value.
11	CHANNEL7	Mirrors values of each SyncManager Status: 0: No Sync Channel 7 event 1: Sync Channel 7 event pending
10	CHANNEL6	Mirrors values of each SyncManager Status: 0: No Sync Channel 6 event 1: Sync Channel 6 event pending
9	CHANNEL5	Mirrors values of each SyncManager Status: 0: No Sync Channel 5 event 1: Sync Channel 5 event pending
8	CHANNEL4	Mirrors values of each SyncManager Status: 0: No Sync Channel 4 event 1: Sync Channel 4 event pending
7	CHANNEL3	Mirrors values of each SyncManager Status: 0: No Sync Channel 3 event 1: Sync Channel 3 event pending
6	CHANNEL2	Mirrors values of each SyncManager Status: 0: No Sync Channel 2 event

		1: Sync Channel 2 event pending
5	CHANNEL1	Mirrors values of each SyncManager Status: 0: No Sync Channel 1 event 1: Sync Channel 1 event pending
4	CHANNEL0	Mirrors values of each SyncManager Status: 0: No Sync Channel 0 event 1: Sync Channel 0 event pending
3	AL_STATUS	AL Status event: 0: No change in AL Status 1: AL Status change (Bit is cleared by reading out AL Status 0x0130:0x0131 from ECAT)
2	DL_STATUS	DL Status event: 0: No change in DL Status 1: DL Status change (Bit is cleared by reading out DL Status 0x0110:0x0111 from ECAT)
1	Reserved	Must be kept at reset value.
0	DC_LATCH	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from ECAT for ECAT-controlled Latch Units, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event)

9.4.34. ESC AL Event Request register (ESC_AL_EVT_REQ)

Address Offset: 0x0220

Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								SYNC15_INT	SYNC14_INT	SYNC13_INT	SYNC12_INT	SYNC11_INT	SYNC10_INT	SYNC9_INT	SYNC8_INT
								r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC7_INT	SYNC6_INT	SYNC5_INT	SYNC4_INT	SYNC3_INT	SYNC2_INT	SYNC1_INT	SYNC0_INT	Reserved	PDATA	EEPROM	SYNC_CHANGE	DC_SYNC1	DC_SYNC0	DC_LATCH	AL_CONTROL
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)		r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)

Bits	**Fields**	**Descriptions**
31:24	Reserved	Must be kept at reset value.

23	SYNC15_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 15 interrupt 1: SyncManager 15 interrupt pending
22	SYNC14_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 14 interrupt 1: SyncManager 14 interrupt pending
21	SYNC13_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 13 interrupt 1: SyncManager 13 interrupt pending
20	SYNC12_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 12 interrupt 1: SyncManager 12 interrupt pending
19	SYNC11_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 11 interrupt 1: SyncManager 11 interrupt pending
18	SYNC10_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 10 interrupt 1: SyncManager 10 interrupt pending
17	SYNC9_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 9 interrupt 1: SyncManager 9 interrupt pending
16	SYNC8_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 8 interrupt 1: SyncManager 8 interrupt pending
15	SYNC7_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 7 interrupt 1: SyncManager 7 interrupt pending
14	SYNC6_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 6 interrupt 1: SyncManager 6 interrupt pending
13	SYNC5_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 5 interrupt 1: SyncManager 5 interrupt pending
12	SYNC4_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 4 interrupt 1: SyncManager 4 interrupt pending
11	SYNC3_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 3 interrupt

		1: SyncManager 3 interrupt pending
10	SYNC2_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 2 interrupt 1: SyncManager 2 interrupt pending
9	SYNC1_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 1 interrupt 1: SyncManager 1 interrupt pending
8	SYNC0_INT	SyncManager interrupts (SyncManager register offset 0x5, bit [0] or [1]): 0: No SyncManager 0 interrupt 1: SyncManager 0 interrupt pending
7	Reserved	Must be kept at reset value.
6	PDATA	Watchdog Process Data: 0: Has not expired 1: Has expired (Bit is cleared by reading Watchdog Status Process Data 0x0440 from PDI)
5	EEPROM	EEPROM Emulation: 0: No command pending 1: EEPROM command pending (Bit is cleared by acknowledging the command in EEPROM Control/Status register 0x0502:0x0503[10:8] from PDI)
4	SYNC_CHANGE	SyncManager activation register (SyncManager register offset 0x6) changed: 0: No change in any SyncManager 1: At least one SyncManager changed (Bit is cleared by reading SyncManager Activation registers 0x0806 etc. from PDI)
3	DC_SYNC1	State of DC SYNC1 (if register 0x0151[7]=1): (Bit is cleared by reading of SYNC1 status 0x098F from PDI, use only in Acknowledge mode)
2	DC_SYNC0	State of DC SYNC0 (if register 0x0151[3]=1): (Bit is cleared by reading SYNC0 status 0x098E from PDI, use only in Acknowledge mode)
1	DC_LATCH	DC Latch event: 0: No change on DC Latch Inputs 1: At least one change on DC Latch Inputs (Bit is cleared by reading DC Latch event times from PDI, so that Latch 0/1 Status 0x09AE:0x09AF indicates no event. Available if Latch Unit is PDI-controlled)
0	AL_CONTROL	AL Control event: 0: No AL Control Register change 1: AL Control Register has been written(AL control event is only generated if PDI emulation is turned off (ESC Configuration 0 register 0x0141[0]=0))

(Bit is cleared by reading AL Control register 0x0120:0x0121 from PDI)

9.4.35. RX Error Port X register (RX_PORTX_ERR) (X = 0,1,2,3)

Address Offset: 0x0300 + X * 2

Reset value: 0x0

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">EEOR_PORTX[7:0]</td><td colspan="8">INVALID_PORTX[7:0]</td></tr>
<tr><td colspan="8">rwc(ECAT)
r(PDI)</td><td colspan="8">rwc(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
15:8	EEOR_PORTX[7:0]	Port X RX Error Counter RX Error counter of Port X (counting is stopped when 0xFF is reached).
7:0	INVALID_PORTX[7:0]	Port X Invalid Frame Counter Invalid frame counter of Port X (counting is stopped when 0xFF is reached).

NOTE: Error Counters 0x0300-0x030B are cleared if one of the implemented RX Error counters 0x0300-0x030B is written (preferably 0x0300). Write value is ignored (write 0). Errors are only counted if the loop of the port is open.

9.4.36. Forwarded RX Error Port X register (FWD_PORTX_ERR) (X = 0,1,2,3)

Address Offset: 0x0308 + X

Reset value: 0x0

This register can be accessed by byte(8-bit).

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">F_ERROR_PORTX[7:0]</td></tr>
<tr><td colspan="8">rwc(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
7:0	F_ERROR_PORTX[7:0]	Forwarded error counter of Port X (counting is stopped when 0xFF is reached).

NOTE: Error Counters 0x0300-0x030B are cleared if one of the implemented RX Error counters 0x0300-0x030B is written (preferably 0x0300). Write value is ignored (write 0). Errors are only counted if the loop of the port is open.

9.4.37. ESC Processing Unit Error register (ESC_PU_ERR)

Address Offset: 0x030C

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0

PU_ERROR[7:0]
rwc(ECAT)
r(PDI)							

Bits	Fields	Descriptions
7:0	PU_ERROR[7:0]	ECAT Processing Unit error counter (counting is stopped when 0xFF is reached). Counts errors of frames passing the Processing Unit.

NOTE: Error Counter 0x030C is cleared if error counter 0x030C is written. Write value is ignored (write 0).

9.4.38. ESC PDI Error Counter register (ESC_PDI_ERR)

Address Offset: 0x030D
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0

PDI_ERROR[7:0]
rwc(ECAT)
r(PDI)							

Bits	Fields	Descriptions
7:0	PDI_ERROR[7:0]	PDI Error counter (counting is stopped when 0xFF is reached). Counts if a PDI access has an interface error.

NOTE: Error Counter 0x030D and Error Code 0x030E:0x030F are cleared if error counter 0x030D is written. Write value is ignored (write 0).

9.4.39. ESC PORT X Lost Link register (ESC_PORTX_LOST_LINK) (X = 0,1,2,3)

Address Offset: 0x0310 + X
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0

LOST_LINK[7:0]
rwc(ECAT)
r(PDI)							

Bits	Fields	Descriptions

7:0	LOST_LINK[7:0]	Lost Link counter of Port X (counting is stopped when 0xff is reached). Counts only if port is open and loop is Auto.

NOTE: Lost Link Counters 0x0310-0x0313 are cleared if one of the implemented Lost Link Counters 0x0310-0x0313 is written (preferably 0x0310). Write value is ignored (write 0).

9.4.40. ESC Watchdog Divider register (ESC_WDG_DIV)

Address Offset: 0x0400

Reset value: 0x09C2

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_DIV[15:0]															

rw(ECAT)

r(PDI)

Bits	Fields	Descriptions
15:0	WATCHDOG_DIV[15:0]	Watchdog divider: Number of 25 MHz tics (minus 2) that represent the basic watchdog increment. (Default value is 100µs = 2498)

9.4.41. ESC Watchdog Time PDI register (ESC_WDG_TIME_PDI)

Address Offset: 0x0410

Reset value: 0x03E8

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_TIME[15:0]															

rw(ECAT)

r(PDI)

Bits	Fields	Descriptions
15:0	WATCHDOG_TIME[15:0]	Watchdog Time PDI: number of basic watchdog increments (Default value with Watchdog divider 100µs means 100ms Watchdog)

NOTE: Watchdog is disabled if Watchdog time is set to 0x0000. Watchdog starts counting again with every PDI access.

9.4.42. ESC Watchdog Time Process Data register (ESC_WDG_TIME_PDATA)

Address Offset: 0x0420

Reset value: 0x03E8

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WATCHDOG_TPD[15:0]															

rw(ECAT)

r(PDI)

Bits	Fields	Descriptions
15:0	WATCHDOG_TPD[15:0]	Watchdog Time Process Data: number of basic watchdog increments (Default value with Watchdog divider 100μs means 100ms Watchdog)

9.4.43. ESC Watchdog Status Process Data register (ESC_WDG_STAT_PDATA)

Address Offset: 0x0440

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															WATCHD OG_STA
															r(ECAT) rw(PDI)

Bits	Fields	Descriptions
15:1	Reserved	Must be kept at reset value.
0	WATCHDOG_STA	Watchdog Status of Process Data (triggered by SyncManagers) 0: Watchdog Process Data expired 1: Watchdog Process Data is active or disabled NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220[6]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220[6]. Writing to this register from PDI is possible; write value is ignored (write 0).

9.4.44. ESC Watchdog Counter Process Data register (ESC_WDG_CNT_PDATA)

Address Offset: 0x0442

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
WATCHDOG_CPD[7:0]							

rwc(ECAT)

r(PDI)

Bits	Fields	Descriptions
7:0	WATCHDOG_CPD[7:0]	Watchdog Counter Process Data (counting is stopped when 0xFF is reached). Counts if Process Data Watchdog expires.

9.4.45. ESC Watchdog Counter PDI register (ESC_WDG_CNT_PDI)

Address Offset: 0x0443
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
WATCHDOG_PDIC[7:0]							

rwc(ECAT)
r(PDI)							

Bits	Fields	Descriptions
7:0	WATCHDOG_PDIC[7:0]	Watchdog PDI counter (counting is stopped when 0xFF is reached). Counts if PDI Watchdog expires.

NOTE: Watchdog Counters 0x0442-0x0443 are cleared if one of the Watchdog Counters 0x0442-0x0443 is written. Write value is ignored (write 0).

9.4.46. ESC EEPOM Configuration register (ESC_EEPROM_CFG)

Address Offset: 0x0500
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						FORCE_ECAT	EEPROM_CTL

						rw(ECAT)
						r(PDI)
							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	FORCE_ECAT	Force ECAT access: 0: Do not change Bit 0x0501[0] 1: Reset Bit 0x0501[0] to 0
0	EEPROM_CTL	EEPROM control is offered to PDI: 0: no 1: yes (PDI has EEPROM control)

9.4.47. ESC EEPOM PDI Access register (ESC_EEPROM_ACC)

Address Offset: 0x0501

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							ACCESS_ EEPROM

							r(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	ACCESS_EEPROM	Access to EEPROM: 0: PDI releases EEPROM access 1: PDI takes EEPROM access (PDI has EEPROM control)

NOTE: write access is only possible if (0x0500[0]=1 or 0x0501[0]=1) and 0x0500[1]=0.

9.4.48. ESC EEPOM Contorl/Status register (ESC_EEPROM_CS)

Address Offset: 0x0502

Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEPROM _IF	ERROR_ WEN	ERROR_A CK	EEPROM_ STA	CHECKSU M	COMMAND[2:0]			SELECT	READ_BY TES	EEPROM_ EMU	Reserved				ECAT_EN
r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) rw(PDI)	r(ECAT) r(PDI)	r(ECAT) rw(PDI)	rw(ECAT) rw(PDI)			r(ECAT) r(PDI)	r(ECAT) r(PDI)	r(ECAT) r(PDI)					rw(ECAT) r(PDI)

Bits	Fields	Descriptions
15	EEPROM_IF	Busy(default 0): 0: EEPROM Interface is idle 1: EEPROM Interface is busy
14	ERROR_WEN	Error Write Enable(default 0): 0: No error 1: Write Command without Write enable NOTE: Error bits are cleared by writing “000” (or any valid command) to Command Register Bits [10:8].
13	ERROR_ACK	Error Acknowledge/Command(default 0): 0: No error

		1: Missing EEPROM acknowledge or invalid command EEPROM emulation only: PDI writes 1 if a temporary failure has occurred. NOTE: Error bits are cleared by writing “000” (or any valid command) to Command Register Bits [10:8].
12	EEPROM_STA	EEPROM loading status(default 0): 0: EEPROM loaded, device information ok 1: EEPROM not loaded, device information not available (EEPROM loading in progress or finished with a failure)
11	CHECKSUM	Checksum Error in ESC Configuration Area(default 0): 0: Checksum ok 1: Checksum error EEPROM emulation for ESC only: PDI writes 1 if a CRC failure has occurred for a reload command.
10:8	COMMAND[2:0]	Command register(default 0): Write: Initiate command. Read: Currently executed command Commands: 000: No command/EEPROM idle (clear error bits) 001: Read 010: Write 100: Reload Others: Reserved/invalid commands (do not issue) EEPROM emulation only: after execution, PDI writes command value to indicate operation is ready. NOTE: Write Enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (EEPROM Busy ends). Writing “000” to the command register will also clear the error bits [14:13]. Command bits [10:8] are ignored if Error Acknowledge/Command is pending (bit 13).
7	SELECT	Selected EEPROM Algorithm: 0: 1 address byte (1Kbit – 16Kbit EEPROMs) 1: 2 address bytes (32Kbit – 4 Mbit EEPROMs)
6	READ_BYTES	Supported number of EEPROM read bytes(default 0): 0: 4 Bytes 1: 8 Bytes
5	EEPROM_EMU	EEPROM emulation: 0: Normal operation (I²C interface used) 1: PDI emulates EEPROM (I²C not used)
4:1	Reserved	Must be kept at reset value.
0	ECAT_EN	ECAT write enable(default 0): 0: Write requests are disabled

1: Write requests are enabled

This bit is always 1 if PDI has EEPROM control.

NOTE: Write Enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (EEPROM Busy ends). Writing “000” to the command register will also clear the error bits [14:13]. Command bits [10:8] are ignored if Error Acknowledge/Command is pending (bit 13).

9.4.49. ESC EEPROM Address register (ESC_EEPROM_ADDR)

Address Offset: 0x0504

Reset value: 0x0

This register can be accessed by word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">EEPROM_ADDR[31:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
rw(PDI)</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">EEPROM_ADDR[31:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
rw(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
31:0	EEPROM_ADDR[31:0]	EEPROM Address 0: First word (= 16 bit) 1: Second word … Actually used EEPROM Address bits: [9:0]: EEPROM size up to 16 Kbit [17:0]: EEPROM size 32 Kbit – 4 Mbit [31:0]: EEPROM Emulation

NOTE: write access depends upon the assignment of the EEPROM interface (ECAT/PDI). Write access is blocked if EEPROM interface is busy (0x0502[15]=1).

9.4.50. ESC EEPROM Data register (ESC_EEPROM_DATA)

Address Offset: 0x0508

Reset value: 0x0

This register can be accessed by word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">EEPROM_READ[15:0]</td></tr>
<tr><td colspan="16">r(ECAT)
rw(PDI)</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

EERPOM_WRITE[15:0]

rw(ECAT)
rw(PDI)

Bits	Fields	Descriptions
31:16	EEPROM_READ[15:0]	EEPROM Read data (data read from EEPROM, higher bytes)
15:0	EERPOM_WRITE[15:0]	EEPROM Write data (data to be written to EEPROM) or EEPROM Read data (data read from EEPROM, lower bytes)

9.4.51. ESC MII Management Control / Status register (ESC_MII_MNG_CS)

Address Offset: 0x0510

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MII_IF	COMMAND_ERROR	READ_EEOR	Reserved			COMMAND[1:0]		PHY_ADDRP0[4:0]					MII_CONFIG	PDI_CTL	WRITE_EN
r(ECAT) r(PDI)	r(ECAT) r(PDI)	rw(ECAT) rw(PDI)				rw(ECAT) rw(PDI)		r(ECAT) r(PDI)					r(ECAT) r(PDI)	r(ECAT) r(PDI)	rw(ECAT) r(PDI)

Bits	Fields	Descriptions
15	MII_IF	Busy: 0: MII Management Interface is idle 1: MII Management Interface is busy
14	COMMAND_ERROR	Command error: 0: Last Command was successful 1: Invalid command or write command without Write Enable Cleared by executing a valid command or by writing “00” to Command register bits [9:8].
13	READ_EEOR	Read error: 0: No read error 1: Read error occurred (PHY or register not available) Cleared by writing to register 0x0511.
12:10	Reserved	Must be kept at reset value.
8:9	COMMAND[1:0]	Command register: Write: Initiate command. Read: Currently executed command 00: No command/MI idle (clear error bits) 01: Read 10: Write Others: Reserved/invalid command (do not issue)

		NOTE: Write enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (Busy ends). Writing “00” to the command register will also clear the error bits [14:13]. The Command bits are cleared after the command is executed.
7:3	PHY_ADDRP0[4:0]	PHY address of port 0 (this is equal to the PHY address offset, if the PHY addresses are consecutive) IP Core since V3.0.0/3.00c: Translation 0x0512[7] =0: Register 0x0510[7:3] shows PHY address of port 0 Translation 0x0512[7] =1: Register 0x0510[7:3] shows the PHY address which will be used for port 0-3 as requested by 0x0512[4:0] (valid values 0-3)
2	MII_CONFIG	MII link detection and configuration: 0: Disabled for all ports 1: Enabled for at least one MII port, refer to PHY Port Status (0x0518 ff.) for details
1	PDI_CTL	Management Interface can be controlled by PDI (registers 0x0516-0x0517): 0: Only ECAT control 1: PDI control possible
0	WRITE_EN	Write enable: 0: Write disabled 1: Write enabled This bit is always 1 if PDI has MI control. NOTE: Write enable bit 0 is self-clearing at the SOF of the next frame, Command bits [10:8] are self-clearing after the command is executed (Busy ends). Writing “00” to the command register will also clear the error bits [14:13]. The Command bits are cleared after the command is executed.

9.4.52. ESC PHY Address register (ESC_PHY_ADDR)

Address Offset: 0x0512

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
PHY_ADDR	Reserved		TARGET_PHY[4:0]				

rw(ECAT)
rw(PDI)
			rw(ECAT)
			rw(PDI)				

Bits	Fields	Descriptions
7	PHY_ADDR	Target PHY Address translation: 0: Enabled 1: Disabled Refer to 0x0512[4:0] and 0x0510[7:3] for details.

6:5	Reserved	Must be kept at reset value.
4:0	TARGET_PHY[4:0]	Target PHY Address Translation 0x0512[7]=0: 0-3: Target PHY Addresses 0-3 are used to access the PHYs at port 0-3, when the PHY addresses are properly configured 4-31: The configured PHY address of port 0 (PHY address offset) is added to the Target PHY Address values 4-31 when accessing a PHY Translation 0x0512[7]=1: 0-31: Target PHY Addresses is used when accessing a PHY without translation

NOTE: write access depends on assignment of MI (ECAT/PDI). Write access is blocked if Management interface is busy (0x0510[15]=1).

9.4.53. ESC PHY Register Address register (ESC_PHY_REG_ADDR)

Address Offset: 0x0513

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved			PHY_RADDR[4:0]				

			rw(ECAT)
			rw(PDI)				

Bits	Fields	Descriptions
7:5	Reserved	Must be kept at reset value.
4:0	PHY_RADDR[4:0]	Address of PHY Register that shall be read/written

9.4.54. ESC PHY Data register (ESC_PHY_DATA)

Address Offset: 0x0514

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_RWADDR[15:0]															
rw(ECAT) rw(PDI)															

Bits	Fields	Descriptions
15:0	PHY_RWADDR[15:0]	PHY Read/Write Data

9.4.55. MII Management ECAT Access State register (MIIM_ECAT_ACC_STAT)

Address Offset: 0x0516

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							MII_ACCESS

							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	MII_ACCESS	Access to MII management: 0: ECAT enables PDI takeover of MII management interface 1: ECAT claims exclusive access to MII management interface

NOTE: write access is only possible if 0x0517[0]=0.

9.4.56. MII Management ECAT Access State register (MIIM_PDI_ACC_STAT)

Address Offset: 0x0517

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						PDI_ACCESS	MII_ACCESS

						rw(ECAT)
						r(PDI)
							r(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	PDI_ACCESS	Force PDI Access State: 0: Do not change Bit 0x0517[0] 1: Reset Bit 0x0517[0] to 0
0	MII_ACCESS	Access to MII management: 0: ECAT has access to MII management 1: PDI has access to MII management

9.4.57. PHY Port X Status register (PHY_PORTX_STAT) (X = 0,1,2,3)

Address Offset: 0x0518 + X

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved		PHY_UPDATE	PARTNER_ERROR	READ_ERROR	STATUS_ERROR	LINK_STATUS	PHYLINK
		rwc(ECAT)	r(ECAT)	r(ECAT)	rwc(ECAT)	r(ECAT)	r(ECAT)
		rwc(PDI)	r(PDI)	r(PDI)	rwc(PDI)	r(PDI)	r(PDI)

Bits	Fields	Descriptions
7:6	Reserved	Must be kept at reset value.
5	PHY_UPDATE	PHY configuration updated: 0: No update 1: PHY configuration was updated Cleared by writing any value to at least one of the PHY Port X Status registers.
4	PARTNER_ERROR	Link partner error: 0: No error detected 1: Link partner error
3	READ_ERROR	Read error: 0: No read error occurred 1: A read error has occurred Cleared by writing any value to at least one of the PHY Port X Status registers.
2	STATUS_ERROR	Link status error: 0: No error 1: Link error, link inhibited
1	LINK_STATUS	Link status (100 Mbit/s, Full Duplex, Auto negotiation): 0: No link 1: Link detected
0	PHYLINK	Physical link status: 0: No physical link 1: Physical link detected

9.4.58. Logical Start address FMMU X register (FMMUX_LOGIC_ADDR) (X = 0…F)

Address Offset: 0x0600 + X * 0x10

Reset value: 0x0

This register can be accessed by word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">LOGIC_ADDR[31:16]</td></tr>
<tr><td colspan="16">rw(ECAT)
r(PDI)</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">LOGIC_ADDR[15:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
31:0	LOGIC_ADDR[31:0]	Logical start address within the EtherCAT Address Space.

9.4.59. Length FMMU X register (FMMUX_LENGTH) (X = 0…F)

Address Offset: 0x0604 + X * 0x10

Reset value: 0x0

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">LENGTH_FMMU[15:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
15:0	LENGTH_FMMU[15:0]	Offset from the first logical FMMU byte to the last FMMU byte + 1 (e.g., if two bytes are used, then this parameter shall contain 2)

9.4.60. Start bit FMMU X in Logical address space register (FMMUX_STRA_BIT) (X = 0…F)

Address Offset: 0x0606 + X * 0x10

Reset value: 0x0

This register can be accessed by byte(8-bit).

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">Reserved</td><td colspan="3">START_BIT[2:0]</td></tr>
<tr><td colspan="5"></td><td colspan="3">rw(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2:0	START_BIT[2:0]	Logical starting bit that shall be mapped (bits are counted from least significant bit

0 to most significant bit 7)

9.4.61. Stop bit FMMU X in Logical address space register (FMMUX_STOP_BIT) (X = 0…F)

Address Offset: 0x0607 + X * 0x10

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved					STOP_BIT[2:0]		

					rw(ECAT)
					r(PDI)		

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2:0	STOP_BIT[2:0]	Last logical bit that shall be mapped (bits are counted from least significant bit 0 to most significant bit 7)

9.4.62. Physical Start address FMMU X register (FMMUX_ADDR) (X = 0…F)

Address Offset: 0x0608 + X * 0x10

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR[15:0]															
rw(ECAT) r(PDI)															

Bits	Fields	Descriptions
15:0	START_ADDR[15:0]	Physical Start Address (mapped to logical Start address)

9.4.63. Physical Start bit FMMU X register (FMMUX_PSBIT) (X = 0…F)

Address Offset: 0x060A + X * 0x10

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved					PSTART_BIT[2:0]		

					rw(ECAT)
					r(PDI)		

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2:0	PSTART_BIT[2:0]	Physical starting bit as target of logical start bit mapping (bits are counted from least significant bit 0 to most significant bit 7)

9.4.64. Type FMMU X register (FMMUX_TYPE) (X = 0…F)

Address Offset: 0x060B + X * 0x10

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						MAP_WRITE	MAP_READ

						rw(ECAT)
						r(PDI)
							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	MAP_WRITE	0: Ignore mapping for write accesses 1: Use mapping for write accesses
0	MAP_READ	0: Ignore mapping for read accesses 1: Use mapping for read accesses

9.4.65. Active FMMU X register (FMMUX_ACTIVE) (X = 0…F)

Address Offset: 0x060C + X * 0x10

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							ACTIVE_FMMU

							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	ACTIVE_FMMU	0: FMMU deactivated 1: FMMU activated. FMMU checks logically addressed blocks to be mapped

according to configured mapping

9.4.66. Physical Start address SyncManager X register (SMX_START_ADDR) (X = 0…F)

Address Offset: 0x0800 + X * 8

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMSTART_ADDR[15:0]															
rw(ECAT) r(PDI)															

Bits	Fields	Descriptions
15:0	SMSTART_ADDR[15:0]	First byte that will be handled by SyncManager

9.4.67. Length SyncManager X register (SMX_LENGTH) (X = 0…F)

Address Offset: 0x0802 + X * 8

Reset value: 0x0

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LENGTH_SM[15:0]															
rw(ECAT) r(PDI)															

Bits	Fields	Descriptions
15:0	LENGTH_SM[15:0]	Number of bytes assigned to SyncManager (shall be greater than 1, otherwise SyncManager is not activated. If set to 1, only Watchdog Trigger is generated if configured)

9.4.68. Control Register SyncManager X register (SMX_CTL) (X = 0…F)

Address Offset: 0x0804 + X * 8

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved	TRIGGER	AL_EVENT_INT	ECAT_EVENT_INT	DIR[1:0]		OPERA_MODE[1:0]	

	rw(ECAT)
	r(PDI)
		rw(ECAT)
		r(PDI)
			rw(ECAT)
			r(PDI)
				rw(ECAT)
				r(PDI)
						rw(ECAT)
						r(PDI)	

Bits	Fields	Descriptions
7	Reserved	Must be kept at reset value.
6	TRIGGER	Watchdog Trigger Enable: 0: Disabled 1: Enabled
5	AL_EVENT_INT	Interrupt in AL Event Request Register: 0: Disabled 1: Enabled
4	ECAT_EVENT_INT	Interrupt in ECAT Event Request Register: 0: Disabled 1: Enabled
3:2	DIR[1:0]	Direction: 00: Read: ECAT read access, PDI write access. 01: Write: ECAT write access, PDI read access. 10: Reserved 11: Reserved
1:0	OPERA_MODE[1:0]	Operation Mode: 00: Buffered (3 buffer mode) 01: Reserved 10: Mailbox (Single buffer mode) 11: Reserved

9.4.69. Status Register SyncManager X register (SMX_STAT) (X = 0…F)

Address Offset: 0x0805 + X * 8

Reset value: 0x30

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
WRITE_BUFFER	READ_BUFFER	BUFFER_STATUS[1:0]		MAILBOX_STATUS	Reserved	READ_INT	WRITE_INT

r(ECAT)
r(PDI)
	r(ECAT)
	r(PDI)
		r(ECAT)
		r(PDI)
				r(ECAT)
				r(PDI)
						r(ECAT)
						r(PDI)
							r(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7	WRITE_BUFFER	Write buffer in use (opened)
6	READ_BUFFER	Read buffer in use (opened)
5:4	BUFFER_STATUS[1:0]	Buffered mode: buffer status (last written buffer): 00: 1st buffer

		01: 2nd buffer 10: 3rd buffer 11: (no buffer written) Mailbox mode: reserved
3	MAILBOX_STATUS	Mailbox mode: mailbox status: 0: Mailbox empty 1: Mailbox full Buffered mode: reserved
2	Reserved	Must be kept at reset value.
1	READ_INT	Interrupt Read: 1: Interrupt after buffer was completely and successfully read 0: Interrupt cleared after first byte of buffer was written NOTE: This interrupt is signalled to the writing side if enabled in the SM Control register.
0	WRITE_INT	Interrupt Write: 1: Interrupt after buffer was completely and successfully written 0: Interrupt cleared after first byte of buffer was read NOTE: This interrupt is signalled to the reading side if enabled in the SM Control register.

9.4.70. Activate SyncManager X register (SMX_ACTIVE) (X = 0…F)

Address Offset: 0x0806 + X * 8

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
LATCH_PDI	LATCH_ECAT	Reserved				REQUEST	SM_EN

rw(ECAT)
r(PDI)
	rw(ECAT)
	r(PDI)
						rw(ECAT)
						r(PDI)
							r(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7	LATCH_PDI	Latch Event PDI: 0: No 1: Generate Latch events when PDI issues a buffer exchange or when PDI accesses buffer start address
6	LATCH_ECAT	Latch Event ECAT: 0: No 1: Generate Latch event when EtherCAT main issues a buffer exchange
5:2	Reserved	Must be kept at reset value.

1	REQUEST	Repeat Request: A toggle of Repeat Request means that a mailbox retry is needed (primarily used in conjunction with ECAT Read Mailbox)
0	SM_EN	SyncManager Enable/Disable: 0: Disable: Access to Memory without SyncManager control 1: Enable: SyncManager is active and controls Memory area set in configuration NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI in all SyncManagers which have changed activation clears AL Event Request 0x0220[4]. Writing to this register from PDI is not possible.

9.4.71. PDI Control SyncManager X register (SMX_PDI_CTL) (X = 0⋯F)

Address Offset: 0x0807 + X * 8

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						ACK	DEACTIVE_SM

						r(ECAT)
						rw(PDI)
							r(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	ACK	Repeat Ack: If this is set to the same value as that set by Repeat Request, the PDI acknowledges the execution of a previous set Repeat request.
0	DEACTIVE_SM	Deactivate SyncManager: Read: 0: Normal operation, SyncManager activated. 1: SyncManager deactivated and reset. SyncManager locks access to Memory area. Write: 0: Activate SyncManager 1: Request SyncManager deactivation NOTE: Writing 1 is delayed until the end of the frame, which is currently processed.

9.4.72. ESC Receive Time Port 0 register (ESC_RCV_TIMEP0)

Address Offset: 0x0900

Reset value: Undefined

This register can be accessed by half-word(16-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCAL_TIMEP0[23:0]															
r(ECAT)															
r(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								RWLOCAL_TIME[7:0]							
								rw(ECAT)							
								r(PDI)							

Bits	Fields	Descriptions
31:8	LOCAL_TIMEP0[23:0]	Local time at the beginning of the last receive frame containing a write access to register 0x0900.
7:0	RWLOCAL_TIME[7:0]	Write: A write access to register 0x0900 with BWR or FPWR latches the local time at the beginning of the receive frame (start first bit of preamble) at each port. Read: Local time at the beginning of the last receive frame containing a write access to this register. NOTE: FPWR requires an address match for accessing this register like any FPWR command. All write commands with address match will increment the working counter (e.g., APWR), but they will not trigger receive time latching.

NOTE: The time stamps cannot be read in the same frame in which this register was written.

9.4.73. ESC Receive Time Port 1 register (ESC_RCV_TIMEP1)

Address Offset: 0x0904

Reset value: Undefined

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCAL_TIMEP1[31:16]															
r(ECAT)															
r(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCAL_TIMEP1[15:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31:0	LOCAL_TIMEP1[31:0]	Local time at the beginning of a frame (start first bit of preamble) received at port 1 containing a BWR or FPWR to register 0x0900.

9.4.74. ESC Receive Time Port 2 register (ESC_RCV_TIMEP2)

Address Offset: 0x0908

Reset value: Undefined

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCAL_TIMEP2[31:16]															
r(ECAT)															
r(PDI)															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOCAL_TIMEP2[15:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31:0	LOCAL_TIMEP2[31:0]	Local time at the beginning of a frame (start first bit of preamble) received at port 2 containing a BWR or FPWR to register 0x0900.

9.4.75. ESC System Time register (ESC_SYS_TIME)

Address Offset: 0x0910

Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
READ_ACESS[31:0]								WRITE_ACESS[31:0]							
r(ECAT)								w(ECAT)							
r(PDI)								w(PDI)							

Bits	Fields	Descriptions
63:0	READ_ACESS[63:0]	ECAT read access: Local copy of the System Time when the frame passed the reference clock (i.e., including System Time Delay). Time latched at beginning of the frame (Ethernet SOF delimiter) PDI read access: Local copy of the System Time. Time latched when reading first byte (0x0910)
31:0	WRITE_ACESS[31:0]	Write access: Written value will be compared with the local copy of the System time. The result is an input to the time control loop. NOTE: written value will be compared at the end of the frame with the latched (SOF) local copy of the System time if at least the first byte (0x0910) was written.

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT/ on=PDI; ECAT control is common).

9.4.76. ESC Receive time ECAT Processing Unit register (ESC_RCVT_ECAT_PU)

Address Offset: 0x0918
Reset value: Undefined

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
RECVE_TIME[63:0]															

r(ECAT)

r(PDI)

Bits	Fields	Descriptions
63:0	RECVE_TIME[63:0]	Local time at the beginning of a frame (start first bit of preamble) received at the ECAT Processing Unit containing a write access to register 0x0900 NOTE: if port 0 is open, this register reflects the Receive Time Port 0 as a 64 Bit value. Any valid EtherCAT write access to register 0x0900 triggers latching, not only BWR/FPWR commands as with register 0x0900.

9.4.77. ESC System Time Offset register (ESC_SYS_TIME_OFFSET)

Address Offset: 0x0920
Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
OFFSET_TIME[63:0]															

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
63:0	OFFSET_TIME[63:0]	Difference between local time and System Time. Offset is added to the local time.

9.4.78. ESC System Time Delay register (ESC_SYS_TIME_DELAY)

Address Offset: 0x0928
Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DELAY_TIME[31:0]															

rw(ECAT)

rw(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELAY_TIME[31:0]															
rw(ECAT)															
rw(PDI)															

Bits	Fields	Descriptions
31:0	DELAY_TIME[31:0]	Delay between Reference Clock and the ESC

9.4.79. ESC System Time Difference register (ESC_SYS_TIME_DIFF)

Address Offset: 0x092C
Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIME_COMPARE	DIFF_TIME[30:0]														
r(ECAT)	r(ECAT)														
r(PDI)	r(PDI)														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFF_TIME[30:0]															
r(ECAT)															
r(PDI)															

Bits	Fields	Descriptions
31	TIME_COMPARE	0: Local copy of System Time less than received System Time 1: Local copy of System Time greater than or equal to received System Time
30:0	DIFF_TIME[30:0]	Mean difference between local copy of System Time and received System Time values Difference = Received System Time – local copy of System Time

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

9.4.80. ESC Speed Counter Start register (ESC_SPD_CNT_START)

Address Offset: 0x0930
Reset value: 0x1000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	BWIDTH[14:0]														
	rw(ECAT)														
	rw(PDI)														

Bits	Fields	Descriptions
15	Reserved	Must be kept at reset value.
14:0	BWIDTH[14:0]	Bandwidth for adjustment of local copy of System Time (larger values -> smaller bandwidth and smoother adjustment) A write access resets System Time Difference (0x092C:0x092F) and Speed Counter Diff (0x0932:0x0933). Valid values: 0x0080 to 0x3FFF

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT / on=PDI; ECAT control is common).

9.4.81. ESC Speed Counter Diff register (ESC_SPD_CNT_DIFF)

Address Offset: 0x0932

Reset value: 0x0000

This register can be accessed by half-word(16-bit).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIFF_CLOCK[15:0]															

r(ECAT)

r(PDI)

Bits	Fields	Descriptions
15:0	DIFF_CLOCK[15:0]	Representation of the deviation between local clock period and Reference Clock's clock period (representation: two's complement) Range: ± (Speed Counter Start – 0x7F)

NOTE: The clock deviation after System Time Difference has settled at a low value can be calculated as follows:

Deviation = Speed Counter Diff / 5(Speed Counter Start + Speed Counter Diff + 2)(Speed Counter Start - Speed Counter Diff + 2)

9.4.82. ESC System Time Difference Filter Depth register (ESC_SYS_TIME_DIFF_FD)

Address Offset: 0x0934

Reset value: 0x04

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved				FILTER_DAPTH[3:0]			

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
7:4	Reserved	Must be kept at reset value.
3:0	FILTER_DAPTH[3:0]	Filter depth for averaging the received System Time deviation

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT/ on=PDI; ECAT control is common).

9.4.83. ESC Speed Counter Filter Depth register (ESC_SPD_CNT_FD)

Address Offset: 0x0935

Reset value: 0x12

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved				FILTER_DAPTH[3:0]			

				rw(ECAT)
				rw(PDI)			

Bits	Fields	Descriptions
7:4	Reserved	Must be kept at reset value.
3:0	FILTER_DAPTH[3:0]	Filter depth for averaging the clock period deviation

NOTE: Write access to this register depends upon ESC configuration (System Time PDI-controlled off=ECAT/ on=PDI; ECAT control is common).

9.4.84. ESC Cyclic Unit Control register (ESC_CYC_UNIT_CTL)

Address Offset: 0x0980

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved		LATCH_UNIT1	LATCH_UNIT0	Reserved			UNIT_CTL

		rw(ECAT)
		r(PDI)
			rw(ECAT)
			r(PDI)
							rw(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:6	Reserved	Must be kept at reset value.
5	LATCH_UNIT1	Latch In unit 1: 0: ECAT-controlled 1: PDI-controlled

		NOTE: Latch interrupt is routed to ECAT/PDI depending on this setting
4	LATCH_UNIT0	Latch In unit 0: 0: ECAT-controlled 1: PDI-controlled NOTE: Latch interrupt is routed to ECAT/PDI depending on this setting. Always 1 (PDI-controlled) if System Time is PDI-controlled.
3:1	Reserved	Must be kept at reset value.
0	UNIT_CTL	Cyclic Unit and SYNC0 out unit control: 0: ECAT-controlled 1: PDI-controlled

9.4.85. ESC Register Activation register (ESC_REG_ACT)

Address Offset: 0x0981
Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
SYNCSIGNAL	CONFIG	START_TIME	EXTEN	AUTO_ACTIVE	SYNC1	SYNC0	SYNC_OUT

rw(ECAT)
rw(PDI)
	rw(ECAT)
	rw(PDI)
		rw(ECAT)
		rw(PDI)
			rw(ECAT)
			rw(PDI)
				rw(ECAT)
				rw(PDI)
					rw(ECAT)
					rw(PDI)
						rw(ECAT)
						rw(PDI)
							rw(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7	SYNCSIGNAL	SyncSignal debug pulse (Vasily bit): 0: Deactivated 1: Immediately generate one ping only on SYNC0-1 according to 0x0981[2:1 for debugging This bit is self-clearing, always read 0. All pulses are generated at the same time, the cycle time is ignored. The configured pulse length is used.
6	CONFIG	Near future configuration (approx.): 0: ½ DC width future (2^{31} ns or 2^{63} ns) 1: ~2.1 sec. future (2^{31} ns)
5	START_TIME	Start Time plausibility check: 0: Disabled. SyncSignal generation if Start Time is reached. 1: Immediate SyncSignal generation if Start Time is outside near future (see 0x0981[6])
4	EXTEN	Extension of Start Time Cyclic Operation (0x0990:0x0993): 0: No extension 1: Extend 32 bit written Start Time to 64 bit

3	AUTO_ACTIVE	Auto-activation by writing Start Time Cyclic Operation (0x0990:0x0997): 0: Disabled 1: Auto-activation enabled. 0x0981[0] is set automatically after Start Time is written.
2	SYNC1	SYNC1 generation: 0: Deactivated 1: SYNC1 pulse is generated
1	SYNC0	SYNC0 generation: 0: Deactivated 1: SYNC0 pulse is generated
0	SYNC_OUT	Sync Out Unit activation: 0: Deactivated 1: Activated

NOTE: Write to this register depends upon setting of 0x0980[0]

9.4.86. ESC Pulse Length of SyncSignals register (ESC_PULSE_SM)

Address Offset: 0x0982

Reset value: depends on configuration

This register can be accessed by half-word(16-bit).

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">PLEN_SM[15:0]</td></tr>
<tr><td colspan="16">r(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
15:0	PLEN_SM[15:0]	Pulse length of SyncSignals (in Units of 10ns) 0: Acknowledge mode: SyncSignal will be cleared by reading SYNC[1:0] Status register

9.4.87. ESC Activation Status register (ESC_ACT_STAT)

Address Offset: 0x0984

Reset value: 0x0

This register can be accessed by byte(8-bit).

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">Reserved</td><td>START_TIME</td><td>SYNC1_ACTIVE</td><td>SYNC0_ACTIVE</td></tr>
<tr><td colspan="5"></td><td>r(ECAT)
r(PDI)</td><td>r(ECAT)
r(PDI)</td><td>r(ECAT)
r(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2	START_TIME	Start Time Cyclic Operation (0x0990:0x0997) plausibility check result when Sync Out Unit was activated: 0: Start Time was within near future 1: Start Time was out of near future (0x0981[6])
1	SYNC1_ACTIVE	SYNC1 activation state: 0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending
0	SYNC0_ACTIVE	SYNC0 activation state: 0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending

9.4.88. ESC SYNC0 Status register (ESC_SYNC0_STAT)

Address Offset: 0x098E

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							SYNC0_STATE

							r(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	SYNC0_STATE	SYNC0 state for Acknowledge mode. SYNC0 in Acknowledge mode is cleared by reading this register from PDI, use only in Acknowledge mode NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220[2]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220[2]. Writing to this register from PDI is possible; write value is ignored (write 0).

9.4.89. ESC SYNC1 Status register (ESC_SYNC1_STAT)

Address Offset: 0x098F

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved							SYNC1_STATE

							r(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7:1	Reserved	Must be kept at reset value.
0	SYNC1_STATE	SYNC1 state for Acknowledge mode. SYNC1 in Acknowledge mode is cleared by reading this register from PDI, use only in Acknowledge mode NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI clears AL Event Request 0x0220[3]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI clears AL Event Request 0x0220[3]. Writing to this register from PDI is possible; write value is ignored (write 0).

9.4.90. ESC Start Time Cyclic Operation register (ESC_START_TIME_CO)

Address Offset: 0x0990

Reset value: 0x0

This register can be accessed by half-word(16-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
START_TIME[63:0]															
rw(ECAT) rw(PDI)															

Bits	Fields	Descriptions
63:0	START_TIME[63:0]	Write: Start time (System time) of cyclic operation in ns Read: System time of next SYNC0 pulse in ns

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

9.4.91. ESC Next SYNC1 Pulse register (ESC_NEXT_SYNC1_PULSE)

Address Offset: 0x0998

Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0

NEXT_SYNC1[63:0]

r(ECAT)

r(PDI)

Bits	Fields	Descriptions
63:0	NEXT_SYNC1[63:0]	System time of next SYNC1 pulse in ns

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

9.4.92. ESC SYNC0 Cycle Time register (ESC_SYNC0_CYC_TIME)

Address Offset: 0x09A0

Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNC0_CYCLE[31:16]															

rw(ECAT)

rw(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC0_CYCLE[15:0]															

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
31:0	SYNC0_CYCLE[31:0]	Time between two consecutive SYNC0 pulses in ns. 0: Single shot mode, generate only one SYNC0 pulse. NOTE: Write to this register depends upon setting of 0x0980[0]. Minimum value for cyclic operation: 60 [ns].

9.4.93. ESC SYNC1 Cycle Time register (ESC_SYNC1_CYC_TIME)

Address Offset: 0x09A4

Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNC1_CYCLE[31:16]															

rw(ECAT)

rw(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC1_CYCLE [15:0]															

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
31:0	SYNC1_CYCLE[31:0]	Time between SYNC0 pulse and SYNC1 pulse in ns NOTE: Write to this register depends upon setting of 0x0980[0].

9.4.94. ESC Latch0 Control register (ESC_LATCH0_CTL)

Address Offset: 0x09A8

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						LATCH0_NEGATIVE	LATCH0_POSITIVE

						rw(ECAT)
						rw(PDI)
							rw(ECAT)
							rw(PDI)

Bits	Fields	Descriptions
7:2	Reserved	Must be kept at reset value.
1	LATCH0_NEGATIVE	Latch0 negative edge: 0: Continuous Latch active 1: Single event (only first event active)
0	LATCH0_POSITIVE	Latch0 positive edge: 0: Continuous Latch active 1: Single event (only first event active)

NOTE: Write access depends upon setting of 0x0980[4].

9.4.95. ESC Latch1 Control register (ESC_LATCH1_CTL)

Address Offset: 0x09A9

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved						LATCH1_NEGATIVE	LATCH1_POSITIVE

						rw(ECAT)
						rw(PDI)
							rw(ECAT)
							rw(PDI)

Bits	Fields	Descriptions

7:2	Reserved	Must be kept at reset value.
1	LATCH1_NEGATIVE	Latch1 negative edge: 0: Continuous Latch active 1: Single event (only first event active)
0	LATCH1_POSITIVE	Latch1 positive edge: 0: Continuous Latch active 1: Single event (only first event active)

NOTE: Write access depends upon setting of 0x0980[5].

9.4.96. ESC Latch0 Status register (ESC_LATCH0_STAT)

Address Offset: 0x09AE

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0
Reserved					L0PIN_STATE	ELATCH0_NEGATIVE	ELATCH0_POSITIVE

					r(ECAT)
					r(PDI)
						r(ECAT)
						r(PDI)
							r(ECAT)
							r(PDI)

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2	L0PIN_STATE	Latch0 pin state
1	ELATCH0_NEGATIVE	Event Latch0 negative edge. 0: Negative edge not detected or continuous mode 1: Negative edge detected in single event mode only. Flag cleared by reading out Latch0 Time Negative Edge.
0	ELATCH0_POSITIVE	Event Latch0 positive edge. 0: Positive edge not detected or continuous mode 1: Positive edge detected in single event mode only. Flag cleared by reading out Latch0 Time Positive Edge.

9.4.97. ESC Latch1 Status register (ESC_LATCH1_STAT)

Address Offset: 0x09AF

Reset value: 0x0

This register can be accessed by byte(8-bit).

7	6	5	4	3	2	1	0

Reserved	L1PIN_STATE	ELATCH1_NEGATIVE	ELATCH1_POSITIVE

	r(ECAT)
	r(PDI)
		r(ECAT)
		r(PDI)
			r(ECAT)
			r(PDI)

Bits	Fields	Descriptions
7:3	Reserved	Must be kept at reset value.
2	L1PIN_STATE	Latch1 pin state
1	ELATCH1_NEGATIVE	Event Latch1 negative edge. 0: Negative edge not detected or continuous mode 1: Negative edge detected in single event mode only. Flag cleared by reading out Latch1 Time Negative Edge.
0	ELATCH1_POSITIVE	Event Latch1 positive edge. 0: Positive edge not detected or continuous mode 1: Positive edge detected in single event mode only. Flag cleared by reading out Latch1 Time Positive Edge.

9.4.98. ESC Latch0 Time Positive Edge (ESC_LATCH0_TIME_PE)

Address Offset: 0x09B0
Reset value: 0x0

This register can be accessed by word(32-bit).

63 .. .. 31 0
L0_POSITIVE[63:0]

r(ECAT)
rw(PDI)

Bits	Fields	Descriptions
63:0	L0_POSITIVE[63:0]	System time at the positive edge of the Latch0 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[0]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[0]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AE[0] if 0x0980[4]=0. Writing to this register from ECAT is not possible.

9.4.99. ESC Latch0 Time Negative Edge (ESC_LATCH0_TIME_NE)

Address Offset: 0x09B8

Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
L0_NEGATIVE[63:0]															

r(ECAT)

rw(PDI)

Bits	Fields	Descriptions
63:0	L0_NEGATIVE[63:0]	System time at the negative edge of the Latch0 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[1]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[4]=1 clears Latch0 Status 0x09AE[1]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AE[1] if 0x0980[4]=0. Writing to this register from ECAT is not possible.

9.4.100. ESC Latch1 Time Positive Edge (ESC_LATCH1_TIME_PE)

Address Offset: 0x09C0

Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
L1_POSITIVE[63:0]															

r(ECAT)

rw(PDI)

Bits	Fields	Descriptions
63:0	L1_POSITIVE[63:0]	System time at the positive edge of the Latch1 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[0]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[0]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AF[0] if 0x0980[5]=0. Writing to this register from ECAT is not possible.

9.4.101. ESC Latch1 Time Negative Edge (ESC_LATCH1_TIME1_NE)

Address Offset: 0x09C8
Reset value: 0x0

This register can be accessed by word(32-bit).

63 .. .. 31 0

L1_NEGATIVE[63:0]

r(ECAT)
rw(PDI)

Bits	Fields	Descriptions
63:0	L1_NEGATIVE[63:0]	System time at the negative edge of the Latch1 signal. NOTE: PDI register function acknowledge by Write command is disabled: Reading this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[1]. Writing to this register from PDI is not possible. PDI register function acknowledge by Write command is enabled: Writing this register from PDI if 0x0980[5]=1 clears Latch0 Status 0x09AF[1]. Writing to this register from PDI is possible; write value is ignored (write 0).

NOTE: Register bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value. Reading this register from ECAT clears Latch0 Status 0x09AF[1] if 0x0980[5]=0. Writing to this register from ECAT is not possible.

9.4.102. ESC Buffer Change Event Time register (ESC_BUF_CET)

Address Offset: 0x09F0
Reset value: 0x0

This register can be accessed by word(32-bit).

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

EVENT_TIME[31:16]

r(ECAT)
r(PDI)

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

EVENT_TIME[15:0]

r(ECAT)
r(PDI)

Bits	Fields	Descriptions
31:0	EVENT_TIME[31:0]	Local time at the beginning of the frame which causes at least one SyncManager to assert an ECAT event

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

9.4.103. ESC PDI Buffer Start Event Time register (ESC_PDI_BUF_SET)

Address Offset: 0x09F8
Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SEVENT_TIME[31:16]															

r(ECAT)

r(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEVENT_TIME[15:0]															

r(ECAT)

r(PDI)

Bits	Fields	Descriptions
31:0	PDI_SEVENT_TIME[31:0]	Local time when at least one SyncManager asserts a PDI buffer start event

NOTE: Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

9.4.104. ESC PDI Buffer Change Event Time register (ESC_PDI_BUF_CET)

Address Offset: 0x09FC
Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CEVENT_TIME[31:16]															

r(ECAT)

r(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CEVENT_TIME[15:0]															

r(ECAT)

r(PDI)

Bits	Fields	Descriptions

31:0	PDI_CEVENT_TIME[31:0]	Local time when at least one SyncManager asserts a PDI buffer change event **NOTE:** Register bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

9.4.105. ESC Product ID register (ESC_PRODUCT_ID)

Address Offset: 0x0E00
Reset value: 0x0

This register can be accessed by word(32-bit).

63 .. .. 31 0

PRODUCT_ID[63:0]
r(ECAT)
r(PDI)

Bits	Fields	Descriptions
63:0	PRODUCT_ID[63:0]	Product ID

9.4.106. ESC Vendor ID register (ESC_VENDOR_ID)

Address Offset: 0x0E08
Reset value: 0x0

This register can be accessed by word(32-bit).

63 .. .. 31 0

Reserved	VENDOR_ID[31:0]
	r(ECAT)
	r(PDI)

Bits	Fields	Descriptions
63:32	Reserved	Must be kept at reset value.
31:0	VENDOR_ID[31:0]	Vendor ID: [23:0] Company [31:24] Department NOTE: Test Vendor IDs have [31:28]=0xE

9.4.107. ESC Digital I/O Output Data register (ESC_DIG_DATA)

Address Offset: 0x0F00
Reset value: 0x0

This register can be accessed by word(32-bit).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OUTPUT_DATA[31:0]															

rw(ECAT)

r(PDI)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUT_DATA[31:0]															

rw(ECAT)

r(PDI)

Bits	Fields	Descriptions
31:0	OUTPUT_DATA[31:0]	Output Data

NOTE: Register size depends on PDI setting and/or device configuration. This register is bit-writable (using Logical addressing).

9.4.108. ESC General Purpose Outputs register (ESC_GP_OUTPUT)

Address Offset: 0x0F10

Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
GP_OUTPUT[63:0]															

rw(ECAT)

rw(PDI)

Bits	Fields	Descriptions
63:0	GP_OUTPUT [63:0]	General Purpose Output Data

NOTE: Register size depends on PDI setting and/or device configuration

9.4.109. ESC General Purpose Inputs register (ESC_GP_INPUT)

Address Offset: 0x0F18

Reset value: 0x0

This register can be accessed by word(32-bit).

63	..	..	..	..	..	..	..	31	..	..	..	..	..	..	0
GP_INPUT[63:0]															

r(ECAT)

r(PDI)

Bits	Fields	Descriptions
63:0	GP_INPUT[63:0]	General Purpose Input Data

NOTE: Register size depends on PDI setting and/or device configuration

9.4.110. ESC User RAM register (ESC_USER_RAM)

Address Offset: 0x0F80

This register can be accessed by word(32-bit).

<table>
<tr><td>1023</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>..</td><td>0</td></tr>
<tr><td colspan="16">USER_RAM [1023:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
rw(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
----	USER_RAM [1023:0]	Application-specific information(128 Bytes)

9.4.111. ESC PDI Digital I/O Input Data register (ESC_PDI_DATA)

Address Offset: 0x1000

Reset value: undefined

This register can be accessed by word(32-bit).

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="16">INPUT_DATA[31:16]</td></tr>
<tr><td colspan="16">rw(ECAT)
rw(PDI)</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="16">INPUT_DATA[15:0]</td></tr>
<tr><td colspan="16">rw(ECAT)
rw(PDI)</td></tr>
</table>

Bits	Fields	Descriptions
31:0	INPUT_DATA[31:0]	Input Data

NOTE: Process Data RAM is only accessible if EEPROM was correctly loaded (register 0x0110[0] = 1).

Input Data size depends on PDI setting and/or device configuration. Digital I/O Input Data is written into the Process Data RAM at these addresses if a Digital I/O PDI with inputs is configured.

9.4.112. ESC Process Data RAM register (ESC_PDRAM)

Address Offset: 0x1000

Reset value: undefined

This register can be accessed by word(32-bit).

65535	..	..	..	..	..	..	..	..	..	..	..	..	..	..	0
PDRAM[8191:0]															
rw(ECAT) rw(PDI)															

Bits	Fields	Descriptions
----	PDRAM [8191:0]	Process Data RAM

NOTE: Process Data RAM is only accessible if EEPROM was correctly loaded (register 0x0110[0] = 1).

10. Revision history

Table 10-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Nov.8, 2024
1.1	1. Modify certain register reset values. 2. Modify the table mode. 3. Add some descriptions for EXMC. 4. Delete the descriptions related to EFUSE in the document. 5. Optimize the naming of ESC registers.	May.15, 2025

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company according to the laws of the People’s Republic of China and other applicable laws. The Company reserves all rights under such laws and no Intellectual Property Rights are transferred (either wholly or partially) or licensed by the Company (either expressly or impliedly) herein. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

To the maximum extent permitted by applicable law, the Company makes no representations or warranties of any kind, express or implied, with regard to the merchantability and the fitness for a particular purpose of the Product, nor does the Company assume any liability arising out of the application or use of any Product. Any information provided in this document is provided only for reference purposes. It is the sole responsibility of the user of this document to determine whether the Product is suitable and fit for its applications and products planned, and properly design, program, and test the functionality and safety of its applications and products planned using the Product. The Product is designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only, and the Product is not designed or intended for use in (i) safety critical applications such as weapons systems, nuclear facilities, atomic energy controller, combustion controller, aeronautic or aerospace applications, traffic signal instruments, pollution control or hazardous substance management; (ii) life-support systems, other medical equipment or systems (including life support equipment and surgical implants); (iii) automotive applications or environments, including but not limited to applications for active and passive safety of automobiles (regardless of front market or aftermarket), for example, EPS, braking, ADAS (camera/fusion), EMS, TCU, BMS, BSG, TPMS, Airbag, Suspension, DMS, ICMS, Domain, ESC, DCDC, e-clutch, advanced-lighting, etc.. Automobile herein means a vehicle propelled by a self-contained motor, engine or the like, such as, without limitation, cars, trucks, motorcycles, electric cars, and other transportation devices; and/or (iv) other uses where the failure of the device or the Product can reasonably be expected to result in personal injury, death, or severe property or environmental damage (collectively "Unintended Uses"). Customers shall take any and all actions to ensure the Product meets the applicable laws and regulations. The Company is not liable for, in whole or in part, and customers shall hereby release the Company as well as its suppliers and/or distributors from, any claim, damage, or other liability arising from or related to all Unintended Uses of the Product. Customers shall indemnify and hold the Company, and its officers, employees, subsidiaries, affiliates as well as its suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Product.

Information in this document is provided solely in connection with the Product. The Company reserves the right to make changes, corrections, modifications or improvements to this document and the Product described herein at any time without notice. The Company shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Information in this document supersedes and replaces information previously supplied in any prior versions of this document.